NOVEL CONVOLUTION-BASED PROCESSING TECHNIQUES FOR APPLICATION IN CHEMICAL SENSING

by

James Edward Taylor

School of Engineering

University of Warwick

A thesis submitted to the University of Warwick

for the degree of Doctor of Philosophy

September 2010

Contents

List of Figures

List of Tables

Summary

The electronic nose is a device developed to mimic the human olfactory system. Despite raising interest from applications in the field of medicine, quality control, environmental control and security, such devices remain inferior to their biological counterparts. As the biological system is explored further, new discoveries generate new ways of thinking in creating electronic nose devices. This has led to a large variety of sensors and devices, all of which produce data that requires processing. The data are processed to extract information that can be used to classify or quantify the input to the electronic nose. However, as the devices have advanced, the data processing techniques have remained relatively static, refinements of established statistical methods.

Recently, investigation into the phenomenon of nasal chromatography has brought about the development of a new class of electronic nose device; the artificial olfactory mucosa. Taking advantage of a retentive effect, inspired by the aqueous mucous layer covering the olfactory epithelium, this new device produces data whose spatio-temporal properties have not been seen in the field of chemical sensing before. Thus there is a need to develop new processing approaches to obtain the information being produced by these new devices.

In this thesis, a new processing approach is presented, centred on the use of convolution to produce characteristic signals which contain information arising from a sensor space that is separated both spatially and temporally, realised in the form of multiple sensor arrays separated by retentive columns or channels. This combined signal is then used to extract an information rich feature set that can be passed on to classifiers or quantifiers to make practical use of the data.

This method is simulated on data collected during the development of the artificial olfactory mucosa to validate its use, and then applied to several sets of real world data, collected from a variety of devices; from current e-nose technologies to newly developed artificial olfactory mucosa devices. The simulations put the device in very noisy conditions and the processing approach deals well with a high level of noise in most circumstances, its performance only deteriorating in the presence of extremely high levels of sensor drift. However, it is shown that this method not only has validity when dealing with the advanced devices for which it is intended, but also shows an improvement over standard processing approaches when utilised in conjunction with current technologies. Utilising convolution on data collected from current devices, methods are developed where the characteristic signal can be generated internally from a single array, and when applied, produce improvements over standard processing approaches.

Acknowledgements

I would like to thank my academic supervisor Prof. Julian Gardner for allowing my opportunity to study in this field, and for his continual support, guidance and understanding through the course of my PhD project. I would also like to acknowledge the Engineering and Physical Sciences Research Council (EPSRC) for their financial support for the period of the PhD project. I am also thankful to my friends and colleagues for their support and advice throughout the course of this project. I would particularly like to thank the members of the *Sensors Research Laboratory, School of Engineering, University of Warwick* for all their encouraging support.

I would also like to thank my collaborators, Dr. Fauzan Che Harun and Dr. James Covington of the *Sensors Research Laboratory, School of Engineering, University of Warwick* for their work on the dual-channel large array electronic nose, and for allowing access to their data, without which this project could not have happened. My thanks also goes to the members of the International Society of Olfaction and Chemical Sensing (ISOCS), in particular Prof. Krishna Persaud, Dr. Jan Mitrovics and Dr. Udo Weimar for their help in obtaining and permission to use data will allowed the scope of this project to expand.

Finally I would like to thank everyone, family and friends, who have supported me through challenging times, and without whose support this work would not have been possible.

Declaration

The work described in this thesis is entirely original and my own work, except when indicated otherwise.

Parts of this work have been presented at international conferences and published in the scientific literature listed below:

[1] J. W. Gardner and J. E. Taylor, *Novel Convolution Based Signal Processing for a Simplified Artificial Olfactory Mucosa*, Transducers '07/Eurosensors XXI, 2007. Presented on June 10-14 2007 in Lyons, France.

[2] F. K. Che Harun, J. E. Taylor, J. A. Covington and J. W. Gardner, *Dual-Channel Odour Separation Columns with Large Chemosensor Arrays for Advanced Odour Discrimination*, IMCS 12, 2007. Presented on July 13-16 2008 in Ohio, USA.

[3] J. E. Taylor, F. K Che Harun, J. A. Covington and J. W. Gardner, *Applying Convolution-Based Processing Methods to a Dual-Channel, Large Array Artificial Olfactory Mucosa*, ISOEN 13, 2009. Presented on April 15-17 2009 in Brescia, Italy.

[4] J. W. Gardner and J. E. Taylor, "Novel Convolution-Based Signal Processing Techniques for an Artificial Olfactory Mucosa", *IEEE Sensors*, vol. 9, no. 8, pp. 929-935, 2009

[5] F. K. Che Harun, J. E. Taylor, J. A. Covington and J. W. Gardner, "An Electronic Nose Employing Dual-Channel Odour Separation Columns with Large Chemosensor Arrays for Advanced Odour Discrimination", *Sensor and Actuator B: Chemical*, vol. 141, Issue 1, pp. 134-140, 2009

List of Abbreviations

Term	Definition
ANN	Artificial Neural Network
AOM	Artificial Olfactory Mucosa
BD	Baseline Drift
c-nose	Retentive column + electronic nose array (Channel Nose)
DFA	Discriminant Function Analysis
e-nose	Electronic Nose
FFT	Fast-Fourier Transform
ISOCS	International Society of Olfaction and Chemical Sensing
KNN	k-Nearest Neighbour
LDF	Linear Discriminant Function
LSER	Linear Solvation Energy Relationship
MCR	Multiple Correlation and Regression
MOX	Metal Oxide chemical sensor
PC	Principal Component
PCA	Principal Components Analysis
PLS	Partial-Least Squares
PNN	Probabilistic Neural Network
QMB	Quartz Micro-Balance
SAW	Surface Acoustic Wave sensor
SNR	Signal-to-Noise Ratio
SP	Stationary Phase
SV	Sample Variance
μC	Microchannel

List of Selected Symbols

Symbol	Definition
$\psi(x)$	Wavelet Function
$n(t)$	Noise Signal
N	Noise Value
s_p	Pooled Sample Variance
$s(t)$	Sensor Signal
S_k	Sensor Response
$S(f)$	Fourier Transform
$S_k^j(t)$	Pre-processed Signal
$X(t)$	Simulated Sensor Signal
$X_k^j(t)$	Noisy Simulated Sensor Signal
$y_k^{ij}(t)$	Characteristic Signal
Z_j	Discriminant Score

Chapter 1: Introduction to Chemical Sensing

1.1 Introduction

Engineering often seeks to mimic our biological senses. Whether it's to broaden what can be perceived, detecting changes with increasing sensitivity or to provide the ability to perceive in hazardous situations, engineering solutions have provided effective artificial substitutes to sight, hearing and touch. However, the senses of taste and smell have proved to be more challenging. While the detection mechanisms for light, sound and pressure have been well understood for a considerable time [1.1], the biological mechanisms for the detection of chemicals have only begun to be understood relatively recently [1.2].

However, as our understanding of the processes of taste (liquid chemical detection) and olfaction (gaseous chemical detection) become more understood, our ability to create devices to sense chemicals improves also. While identifying individual chemicals is possible, the mixtures encountered in nature can dilute certain chemicals to parts per billion or parts per trillion. While these very small taints become increasingly difficult to detect in the environment, biological systems remain sharply aware of these slight taints. Detecting these small concentrations can be very important, as some airborne chemicals can be lethal to humans and animals at parts per billion. Thus, there are major applications for

detecting taints of chemicals in liquids or gases in the field of safety: monitoring air quality, monitoring food quality, etc. in situ.

This thesis explores some of the advances currently being made in the field on chemical sensing, with a particular focus on the processing and classification process. While our understanding of chemical sensing advances [1.3], and allows the development of new types of chemical sensing devices, the techniques being used to process sensor data have remained fairly standardised [1.4]. New breakthroughs in our understanding of biological chemical sensing have led to devices that are producing types of data that these techniques are unsuitable for. This thesis summarises the devices being used and developed in chemical sensing, and presents a new approach to classifying data from a new generation of chemical sensors.

1.2 The Process of Olfaction

Olfaction is the process commonly referred to as the sense of smell. It is the system through which organisms sense their environment through the sensing of chemicals present in the air [1.5]. The chemicals detected by this process are only present in parts per billion, or even parts per trillion. The organs evolved to perform this function are therefore highly sensitive, but also some of the most primitive. This results in them being more closely connected to the brain and able to cause strong reactions in organisms [1.6].

In mammals, the primary sensing organ in olfaction is the nose, more specifically the olfactory epithelium that lines the upper regions inside the nose's airway, shown in figure 1.1.

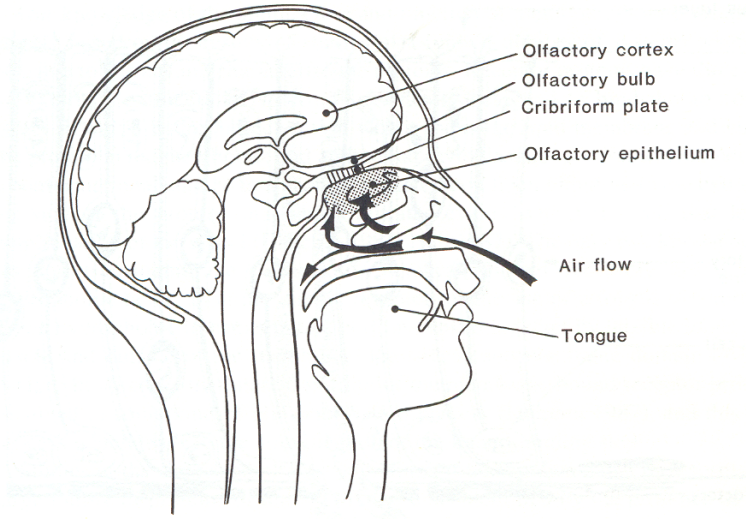


Figure 1.1 The upper airway and location of the olfactory epithelium in humans. [1.7]

Figure 1.2 shows the olfactory epithelium in detail [1.8]. Air passes over this layer, and diffuses through the aqueous mucous layer coating the epithelium. In this mucosa, olfactory cilia sense the chemicals diffused in the layer, and pass the signal onto the brain to be processed and identified. Different regions of the epithelium react differently to differing chemicals and/or concentrations of chemicals. This combination of signals from the epithelium allows the brain to interpret the chemicals in the air as a smell.

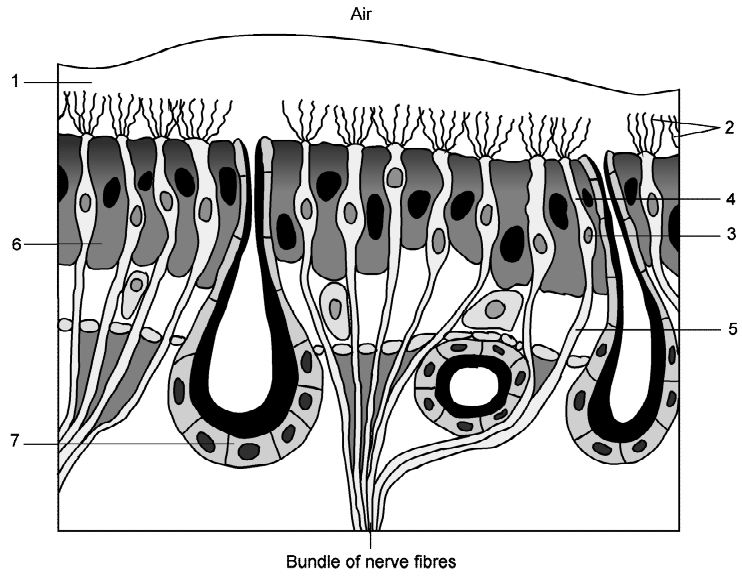


Figure 1.2 The olfactory epithelium. Located in the mucous layer (1) are the olfactory cilia (2) of the sensory cells. These cells consist of a cellular body (3), a dendrite (4) and a neurite or axon (5). They are wedged in between support cells (6). The mucous layer is maintained by Bowman's glands (7). [1.8]

The sensation of smell is the least understood, and the most complex. While light, sound and pressure are well-defined sensations, with well-documented relations between wavelength, magnitude and the sensation produced as a result, such a defined set of reactions and sensations is difficult to find. The sensation of smell is difficult to characterise, with some sources suggesting a few 'basic' smells [1.7], while other sources can produce a list of many basic sensations, such as that shown in Table 1.1. [1.9].

Aldehydic	Camphor	Fresh	Linalool	Resins
Amber	Cinnamic	Fruity	Minty	Seeds
Animalic	Citrus	Green	Mossy	Spicy
Anisic	Floral	Herbal	Musk	Sweet
Aromatic	Floral Fresh	Iris	Patchouli	Watermelon
Balsamic	Floral	Lavender	Pine	Woody

Table 1.1 A list of basic smell sensations used in perfumery [1.9]

This problem is further compounded by the fact that the simple chemical composition does not accurately match up with the perception of smell either [1.10]. Chemically identical isomers, such as those in Figure 1.3, have different smells. Thus, a pure chemical analysis is unable to categorise the sense of smell either.

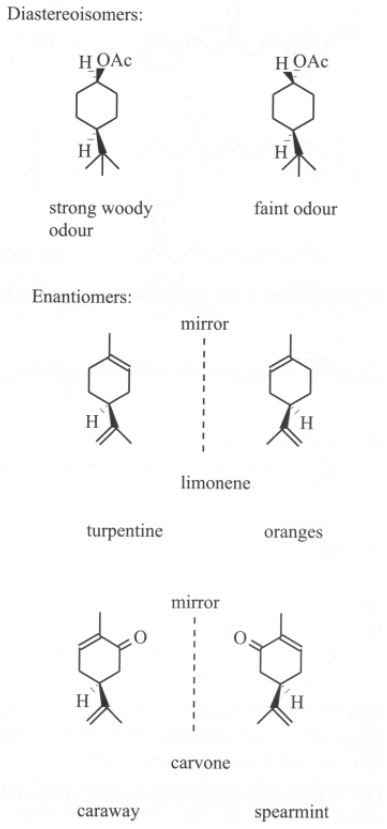


Figure 1.3 Examples of diastereoisomers and enantiomers, whose different structures produce noticeably differing odours. [1.10]

These problems have led to great interest in developing systems to mimic olfactory behaviour, above and beyond the need for simple chemical sensing. For example, while chemical sensing can be used to detect hazardous or lethal substances in air, artificial olfaction can be used to maintain pleasing air quality.

1.3 Chemical Sensing and Artificial Olfaction

The devices used in the field of chemical sensing can be categorised as one of two types: chemical sensors and artificial olfactory devices [1.7]. Chemical sensors are discrete devices, composed of a material whose electrical, optical or acoustic properties change when in the presence of a chemical. Ideally, such sensors would be only sensitive to a single chemical, but this is rarely the case. Many are sensitive to a range of chemicals [1.11]. Artificial olfactory devices, commonly referred to as electronic noses, utilise multiple chemical sensors, each tuned to a different set of chemicals, producing an array of partially specific sensing elements [1.7]. These devices also may consist of other additional devices, such as pre-concentrators and classification systems to produce a complete device.

1.3.1 Types of Chemical Sensor

Chemical sensors are the basic discrete sensing element used in the field of chemical sensing and artificial olfaction. There are several different types of

chemical sensor, differing in construction, sensitivity, ideal environment, etc. Outlined below are the major categories of chemical sensors.

1.3.1.1 Metal Oxide Sensors

Metal oxide sensors consist of a metal oxide material deposited between electrodes. As the chemical is adsorbed and reacts with the oxygen in the oxide, the resistivity of the sensor alters and can be measured [1.7]. Some examples of metal oxide sensors are shown in Figure 1.4.

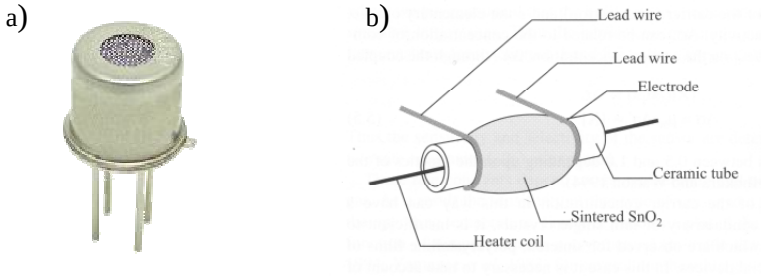


Figure 1.4 Examples of Metal Oxide Sensors: (a) TGS2620 Commercial Alcohol Vapour Metal Oxide Sensor – Figaro Engineering Inc. [1.11] (b) Taguchi-type tin dioxide sensor [1.7]

Such sensors operate ideally at elevated temperatures (300-400°C), and require heating elements which may interfere with other systems or environmental control, as well as increasing power usage. These sensors are very stable, and a properly designed system can remove temperature as an environmental variable [1.7].

1.3.1.2 Polymer Sensors

Polymer sensors consist of an electroactive conducting polymer, or a non-conducting polymer material mixed with carbon black micro-beads, deposited between electrodes. Absorption of a chemical alters the conductive properties of the polymer, or as polymer mixture absorbs a chemical, it expands, increasing the spacing between the carbon black within the polymer, altering the resistance of the polymer. This change in resistivity can then be measured [1.7]. Examples of polymer sensors are shown in figure 1.5.

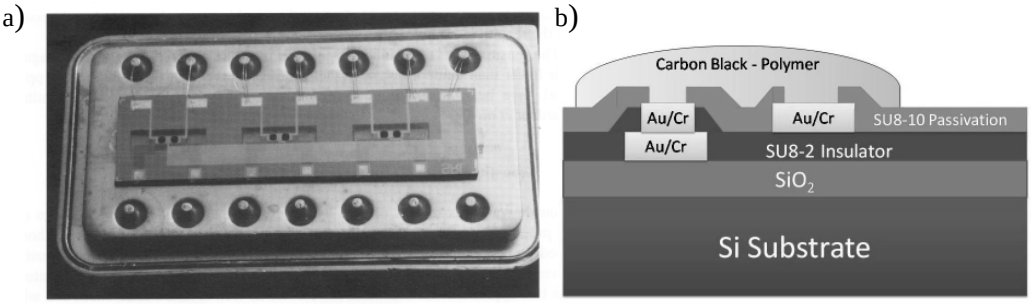


Figure 1.5 Examples of polymer sensors: (a) Electroactive conducting polymer sensors on a silicon chip [1.7] Structure of a Carbon-black polymer sensor [1.12].

Polymer sensors operate at room temperature, have a relatively rapid absorption/desorption cycle and can easily be integrated with silicon technologies. However, they have a low stability, a high sensitivity to environmental factors and temperature, and can be easily poisoned (introduced to a chemical that permanently bonds to the polymer). Polymer sensors have a high sensitivity to organic chemicals but can generally be designed to be sensitive to most compounds. Their selectivity is non-specific however, resulting in a partial sensitivity to a broad range of chemicals [1.7] [1.12].

1.3.1.3 Acoustic Sensors

Acoustic sensors detect analytes through a mechanism which involves altering a vibrational frequency within the sensor. Quartz micro-balance sensors consist of a gold plate covered in an absorption material, placed in contact with a quartz crystal. As the polymer absorbs a chemical, its mass alters and changes the density of the crystal, which alters its resonant frequency. This change in frequency is then measured.

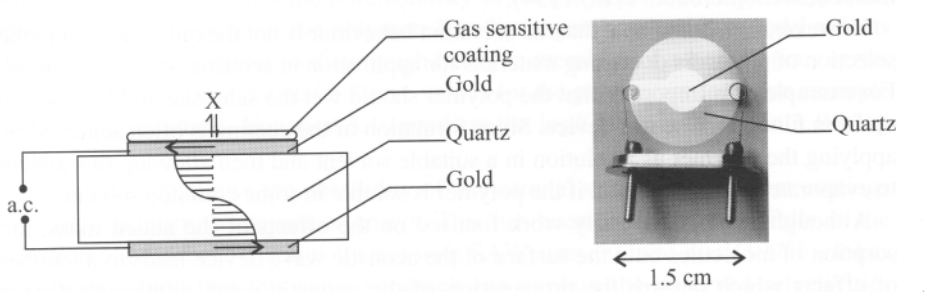


Figure 1.6 Structure of a quartz micro-balance sensor [1.7]

Quartz micro-balance sensors utilise a frequency measurement rather than resistance measurement, reducing the impact of many environmental noise factors such as temperature. The sensitivity of such devices is largely dependant on the materials used in the balance, but require careful calibration and very good humidity control to produce best results [1.7] [1.13].

Surface Acoustic Wave sensors (or SAWs) detect a change in wave frequency when an analyte rests on the sensor's surface. These are typically used to detect chemicals dissolved in liquid. This can either be a gaseous odour dissolved in a

solvent or using a liquid as the analyte itself (for an 'artificial tongue' device) [1.13].

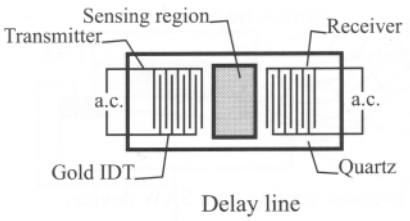


Figure 1.7 Structure of a delay line type SAW sensor [1.7]

1.3.2 Artificial Olfaction

While the above sensors can be used to detect certain chemicals in the air at suitable concentrations, on their own they cannot be used to identify odours. Instead, groups of sensors are used in conjunction with other systems to create an artificial olfactory device. The sensors are created with several differing partially specific tunings. The sensors then produce an array of data, which can form a pattern that can be used to identify an odour in an analytical manner.

1.3.2.1 Electronic Noses

Electronic noses are the simplest type of artificial olfactory device [1.7]. An array of various chemical sensor types, each tuned in a different manner so as to respond differently as odours are presented to the system. These devices need to balance the need for multiple sensor tunings to identify different analytes with reducing possible redundancy and keeping processing time to a minimum. Such sensors are prone to all the problems outlined above with chemical sensors,

although increasing the number of each type of sensor can help to mitigate some of these problems. Environmental noise, poisoning and sensor drift still remain as major problems [1.14].

More advanced electronic noses use multiple arrays of parallel sensors, as well as multiple types of sensor construction to improve accuracy. The addition of sampling systems and pre-concentrators also help to boost the sensitivity of such devices [1.13].

Figure 1.8 Examples of Electronic Noses Systems: Alpha MOS Fox with autosampler [1.15] and Cyrano Sciences C320 (now Smith Detection) [1.16]

1.3.2.2 Gas Chromatography

By combining electronic noses with other chemical devices, more advanced systems can be created. By combining an uncoated SAW device with gas chromatography, a device known as the z-nose™ [1.17] is created. This uses the fact that gases diffuse at different rates through a medium – each differing

chemical travels through the medium at different speeds, each arriving in turn at the sensors. This helps to isolate and clarify the different components and elements that a given gas or odour is composed of. However, the chromatography step can extend the time required to detect samples considerably. Selectivity and discriminatory power can be sacrificed by using a short column to speed up the process.

1.3.2.3 Mass Spectrometry

Mass spectrometers can also be used to identify the chemicals that form an odour [1.18]. Such devices can accurately measure the composition, but as outlined above, some isomers have different odours, making this method a good chemical sensor, but less than ideal olfaction device. Mass spectrometers are also large devices, not suitable for portable or customer-level applications [1.19].

1.4 Processing Chemical Sensor Data

Once collected from the various types of sensors, the data need to be processed so that they can be understood. While there are several methods for doing this, the overall flow of the processing process is composed of a few identifiable steps [1.20]. First the data are subjected to pre-processing, a processing step where the data are prepared for the main processing technique to be carried out. Following this, features of the data are selected and extracted from the data. These features are chosen based on the data and the processing techniques being used, and

should highlight the properties of the data useful for the desired application. This extracted feature set is then subjected to the main processing step. After the processing is complete, some post-processing of the data can be further performed, to make the processed data more suitable for use in classification or presentation. Classification is often the goal of data processing, particularly in the field of chemical sensing and artificial olfaction. Like processing, there are several methods of classification, tailored to certain types of problems and applications. Figure 1.9 outlines the overall flow of a processing operation.

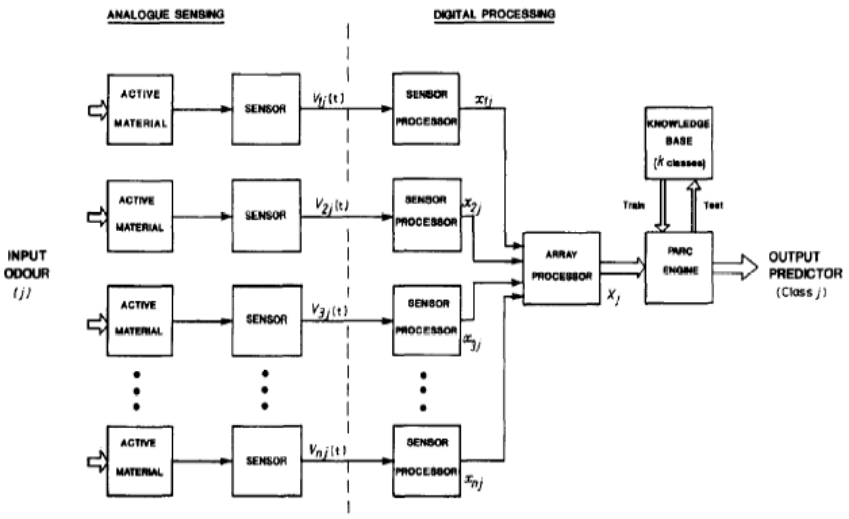


Figure 1.9 Processing and classification flow for processing data from chemical sensing systems [1.21].

1.4.1 Processing Techniques

The processing stage consists of several major steps: pre-processing, processing, feature selection, feature extraction and post-processing [1.22]. Each of these stages utilise several different techniques to improve the quality of the data and

to make the data suitable for use afterwards, such as for presentation or classification. The methods at each stage are carefully chosen based on the nature of the data to be processed. Some data may require very little processing, and just require preparation for use, while other data need much pre- and post-processing to remove influence from factors such as noise, bias and inconsistent results.

Processing aims to take a large amount of data and reduce it to a few key features highlighting the major information contained within the data. However, suitably sized data sets are not always available, particularly in the field of chemical sensing. As such, more advanced techniques are needed for processing these smaller data sets while still maintaining correctness and accuracy.

Pre-processing is the first processing step. The aim of this step to remove the most noticeable influences on the data: noise and bias. Filters can be used to remove noise from signals, and the data can be normalised to remove magnitude bias from the data. However, the use of these techniques comes at a cost, both add processing time, filters need to be carefully chosen so that the data are not destroyed and normalisation magnifies noise on smaller signals [1.23].

Processing is the main operation carried out on the data. A good processing method transforms the data in a manner to highlight information of interest. Processing methods such as correlation and convolution compare signals to bring out extra information. Windowing and filtering ensure the correct segments of data are used to provide information. Averaging multiple signals takes advantage

of in-built redundancy to remove sensor variations and noise sources and focus on a mean signal.

Feature selection and extraction takes the processed data signals, and extracts the key elements of information from the signal for application. A signal has many features suitable for this process, such as peak magnitude, gradient, range, area, decay time, etc. It is important to select the features that make best use of the processed data, even using arithmetic combinations of multiple features. Once a feature, or combination of features, has been selected, these are then extracted to the final processed feature array.

Occasionally, an additional post-processing step may be required to make the processed feature array suitable for use in its chosen application, such as further normalisation of the feature data for use in classification systems or neural networks [1.24].

1.4.2 Classification Techniques

Once data have been processed, it can be used in other applications. In chemical sensing, this application is most often classification of the chemical. The processing is needed to increase the distinction between groups of chemicals that have sensor signals that appear superficially similar. The processing highlights the differences in these signals, but a classifier is needed to group the signals based on these differences. There are two main approaches to classification: statistical grouping and non-parametric classification.

1.4.2.1 Statistical Techniques

Statistical classifiers take data and form groups based purely on the statistical properties of the data [1.25]. Some methods, such as principal components analysis and self-organising maps, simply attempt to maximise the difference between groups in the data, but without any *a priori* information regarding the number or size of groups. Others, such as nearest-neighbour groups and linear discriminant functions form groups based on *a priori* information about the number of groups. Once the analysis method has been applied, groups can be divided and classified by a regression method and relevant identifiers applied to the groups.

The statistical nature of these methods make them favoured, due to the ease with which the information produced by them can be interpreted and presented, particularly if the number of significant dimensions can be reduced to 2 or 3. However, outliers in the data reduce the accuracy of such systems, and the identifier is unable to classify new input classes without a re-calibration of the classification system and groups.

Principal Components Analysis

Principal Components Analysis (PCA) transforms data so that they are redefined in terms of their principal components (PCs). These components are orthogonal

vectors which best describe the variance of the data. The first PC contains the largest amount of variance, followed by successive PCs [1.26].

To calculate PCs, the mean of each sensor's data is set to zero (autoscaled) by subtracting the empirical mean for that sensor from each observation. The covariance of the entire data set is then found. The eigenvectors of this adjusted covariance matrix represent the PCs of the data set, with the eigenvalues representing the amount of variance contained within each PC/eigenvector. The first PC is that with the highest variance/eigenvalue, the second PC is that with the second highest eigenvalue, and so on.

These PCs can then be used as bases for representing the data. Each PC score is generated by multiplying a set of observations by the PC function (Equation 1.1).

$$Score_i = \sum a_i S_i \tag{1.1}$$

where a_i are the PC coefficients and S_i are the individual observations. Scores can be generated for each PC. If a significant amount of variance (>80%) is contained within the first few PCs (2 or 3), these scores can be plotted against each other, and give a good representation of any groupings present within the data. [1.26]

If the information is intended for use in a classification system, multiple linear regression can be used to determine boundary conditions.

Additional information, more suitable for comparing different pre-processing methods, can be extracted from a PCA. Firstly, the amount of variance can be used to judge how well any groups are being represented. Secondly, PC coefficients can be plotted against each other for each sensor. Groups of sensors represent co-sensitivity – where multiple sensors do not contribute information not already conveyed by another in the group. Such groups often introduce additional unwanted noise to the analysis, and can be used in feature selection to further improve a system.

Discriminant Function Analysis

Discriminant Function Analysis (DFA), in a similar manner to PCA, also transforms data using linear discriminant functions (LDFs). However, where PCA attempts to maximise all variances present in the data, DFA generates an LDF to maximise the distance between two given centriods within the data, taking the variances of the two groups into account. So, where PCA could be seen as part of an unsupervised classification system, DFA forms part of a supervised classification system – groups must be known *a priori* to perform DFA [1.27].

LDFs are obtained for a data set using the UNISTAT 5.5 statistical software package. The functions are then exported for use in MATLAB 7.4.0 (R2007a).

A discriminant score for each set of observations, Z_i, can then be determined by multiplying the LDF by the observations (equation 1.2).

$$Z_i = \sum b_i S_i \tag{1.2}$$

where b_i are the LDF coefficients and S_i are the individual observations. Classification is made by proximity to the centroids once converted to this 1-dimensional Z domain [1.28].

A measure of confidence in the LDF can be calculated using the Z scores in a Fisher's t-test [1.27].

$$t_0 = \frac{\overline{Z_j} - \overline{Z_k}}{s_p \sqrt{\frac{1}{n_j} + \frac{1}{n_k}}} \tag{1.3}$$

where $\overline{Z_j}$ and $\overline{Z_k}$ are the sample means of the Z scores for the two groups, *j* and *k*, n_j and n_k are the number of members of the two groups, and s_p is the pooled sample variance, given in equation 1.4 [1.27].

$${s_p}^2 = \frac{(n_j - 1){s_j}^2 + (n_k - 1){s_k}^2}{n_j + n_k - 2} \tag{1.4}$$

As with PCA, there are as many LDFs as there are dimensions in the data. The two most significant LDFs are selected to produce a 2-D plot to represent the data. While the first LDF is usually sufficient for classification, this additional dimension makes the data more suitable for graphical comparisons with other

methods (such as PCA). This also allows for the same co-sensitivity analysis as described for PCA, although LDFs are first normalised so that the maximum length within a group is 1.

1.4.2.2 Non-parametric Techniques

Non-parametric approaches to classification, characterised by neural networks and fuzzy logic systems, rely less on the manipulation of statistical differences in the data and focus on the actual statistical properties themselves [1.29]. These systems classify data based on properties, assigning data to a group based on the combination of the various properties the data possesses. This produces a flexible system that is more responsive to new classes of input, able to detect when an input does not belong to any of the already established classes. However, implementing such systems is complex, and they require additional data so that the system can be trained to recognise input.

Probabilistic Neural Network

Artificial neural networks are powerful classification tools [1.30]. Rather than making direction comparisons between a data sample and a specific set of criteria, data are compared to categories of criteria, and grouped with the most appropriate set of criteria. A typical artificial neural network is illustrated in Figure 1.10.

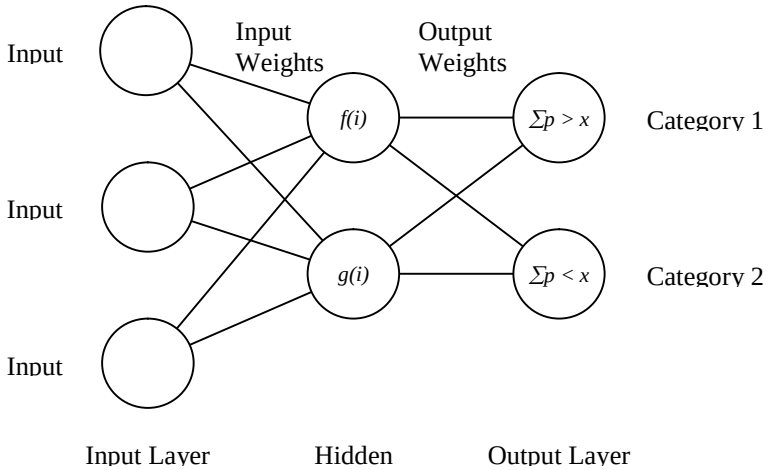


Figure 1.10 A typical neural network structure (a multilayer perceptron after training).

Neural networks are defined in the manner in which they perform classification within each neurone, the number of neurone layers and the learning paradigms used during the training process [1.31]. For example, in the simple network shown in Figure 1.10, an input vector of 3 values is given to the network, each input value is weighted and passed to the processing neurones in the hidden layer. These neurones then perform processing on the weighted input values (*f(i)* and *g(i)*). This processing can be performed in several ways, such as thresholding values, probability functions or fuzzy categories. The results are then weighted, and summed in the output node, and a comparison made – in this case, to a static value *x*. The input vector is then assigned to a category based on the result of the comparisons made in the output neurones.

A probabilistic neural network (PNN) is a feed-forward network with a single hidden layer of neurones. Within each neurone, the input sample is compared to

a set of values and assigned a probability of being similar to that set. This is performed within each neurone in the hidden layer. The input data sample is assigned a classification group based on the combination of probabilities across all the neurones [1.32].

1.5 Summary

This chapter has outlined the evolving field of chemical sensing. The many and varied branches of research each carry significant applications in industry and the world at large. The complex interconnection of sensors, systems, processing and classification produces many different approaches to solve any chemical sensing problem, each carrying its own advantages and disadvantages. With so many ways of combining technologies and methods, novel devices and approaches are being constantly developed within this field. These new devices and approaches create new categories and types of data, which in turn require new techniques to process and make best use of the data being produced.

This thesis focuses on a new processing solution for novel artificial olfactory devices currently in development at the Sensors Research Laboratory, University of Warwick and their partners. These devices require a new approach to data processing, as current techniques used in the field of chemical sensing cannot fully exploit the information being produced. This novel processing technique is centred on the concept of the combination of spatially separated signals,

exploiting the additional information held in the differences between signals, as much as the information held in the signals themselves.

1.6 References

[1.1] H. R. Schiffman, *Sensation and Perception: An Integrated Approach 4th Ed.*, New York; Chichester: John Wiley & Sons, 1996

[1.2] S. F. Takagi, *Human Olfaction*, Tokyo: University of Tokyo Press, 1989

[1.3] M. J. Madou and S. R. Morrison, *Chemical Sensing with Solid State Devices*, Academic Press, 1989

[1.4] J. W. Gardner and P. N. Bartlett, "Performance Definition and Standarization of Electronic Noses", *Sensors & Actuators B*, vol. 33 (1996), pp. 60-67

[1.5] M. J. Serby and K. L. Chobor, *Science of Olfaction*, New York; London: Springer-Verlag, 1992

[1.6] P. A. Vroon, *Smell: The Secret Seducer*, New York: Farrar, Straus and Giroux, 1994

[1.7] J. W. Gardner and P. N. Bartlett, *Electronic Noses: Principles and Applications*, Oxford: Oxford University Press, 1999

[1.8] D. F. Proctor and I. Andersen, *The Nose: Upper Airway Physiology and the Atmospheric Environment*, Amsterdam; New York: Elsevier Biomedical Press, 1982

[1.9] R. R. Calkin and J. S. Jellinek, *Perfumery: Practice and Principle*, New York: John Wiley & Sons, 1994

[1.10] G. Ohloff, *Scent and Fragrances*, Berlin; London: Springer-Verlag, 1994

[1.11] Figaro Engineering Inc., internet source (last accessed 21/03/2010) http://www.figaro.co.jp/

[1.12] F. K. Che Harun, "Mimicking the Human Olfactory System: A Portable e-Mucosa", Ph.D. dissertation, School of Engineering, University of Warwick, Coventry, UK, 2009

[1.13] T. C. Pierce, S. S. Schiffman, H. T. Nagel and J. W. Gardner, *Handbook of Machine Olfaction: Electronic Nose Technology*, Weinheim: Wiley-VCH, 2003

[1.14] S. V. Vaseghi, *Advanced Signal Processing and Digital Noise Reduction*, Chichester: Wiley/Teubner, 1997

[1.15] Alpha M.O.S., internet source (last accessed 21/03/2010) http://www.alpha-mos.com/products_technology/electronic_nose.html

[1.16] Cyrano Sciences Inc., internet source (last accessed 21/03/2006) – now Smiths Detection http://www.smithsdetection.com/

[1.17] A. K. Wanekaya *et al.*, "Multicomponent analysis of alcohol vapors using integrated gas chromatography with sensor arrays", *Sensors and Actuators B*, vol. 110, pp. 41-48, 2005

[1.18] C. A. McDowell, *Mass Spectrometry*, New York; London: McGraw-Hill, 1963

[1.19] J. Barker, *Mass Spectrometry 2nd Edition*, New York: John Wiley & Sons, 1999

[1.20] N. A. Hamdy, *Applied Signal Processing: Concepts, Circuits and Systems*, Boca Raton: CRC Press/Taylor & Francis, 2009

[1.21] J. W. Gardner and P. N. Bartlett, "A Brief History of Electronic Noses", *Sensors and Actuators B*, vol. 18 (1994), pp. 211-210

[1.22] H. Baher, *Analog & Digital Signal Processing*, Chichester: John Wiley & Sons, 2001

[1.23] L. Tan, *Digital Signal Processing*, Amsterdam; Boston: Academic Press, 2008

[1.24] S. Banks, *Signal Processing, Image Processing and Pattern Recognition*, London: Prentice Hall, 1990

[1.25] A. Webb, *Statistical Pattern Recognition*, New York: John Wiley & Sons, 2002

[1.26] A. A. Afifi and V. Clark, *Computer-Aided Multivariate Analysis 3rd Edition*, London: Chapman & Hall 1996

[1.27] A. A. Afifi and S. P. Azen, *Statistical Analysis: A Computer Oriented Approach 2nd Edition*, New York; London: Academic Press 1979

[1.28] C. J. Huberty, *Applied Discriminant Analysis*, New York; Chichester: John Wiley & Sons, 1994

[1.29] B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, 1996

[1.30] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Upper Saddle River, N.J.: Prentice Hall, 1999

[1.31] P. D. Wasserman, *Advanced Methods in Neural Computing*, New York: Van Nostrand Reinhold, 1993

[1.32] F-L. Luo and R. Unbehauen, *Applied Neural Networks for Signal Processing*, Cambridge: Cambridge University Press, 1997

Chapter 2: Review: Time-Dependent Techniques in Chemical Sensing

2.1 Introduction

This thesis aims to introduce a new methodology for processing chemical sensor data. The established techniques described in Chapter 1 all possess one universal quality; they are time-independent approaches to data processing and classification. The research into nasal chromatography [2.1] [2.2] strongly suggests that time plays a role in the identification of odour: the differing times it takes odours to cross the mucosa, the rates at which the concentration of odour changes on the epithelium, and the time taken for odours to be carried away. The time-independent techniques consider these factors in isolation from each other, and in relation to spatial information in the form of concentrations. This temporal information is not considered in relation to other temporal information an odour may be supplying. Thus, this thesis aims to address this untapped source of information by utilising a time-dependent processing approach to make use of this spatio-temporal data set that is being observed by devices developed to make use of the nasal chromatography effect [2.3]. This chapter gives an overview of some of the time-dependent processing techniques that are being applied to chemical sensor data.

2.2 Time-dependent Processing Techniques

Where time-independent processing techniques focus on manipulating the data in the time domain, and treat time as an independent variable within its processing calculations, time-dependant processes transform the data into some other domain. This process makes time a dependent variable in this new transform domain [2.4].

The advantage of doing this is to give another perspective to the data through the addition of the time variable. By converting the temporal factor into a meaningful element of the data, more information can be extracted from the data which could not be considered in the time domain alone. In the following sections, two major forms of time-dependent transforms are suggested, and their current uses in chemical sensing explored.

2.2.1 Fast-Fourier Transform

The Fast-Fourier Transform (FFT) is an evolution of the basic Fourier transform. The Fourier transform decomposes a signal into a series of sinusoids of varying frequencies and amplitudes, which when superpositioned recreate the original signal. This is expressed in equation 2.1.

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt \qquad (2.1)$$

where *s(t)* is the original signal, *S(f)* is the Fourier transform and $j = \sqrt{-1}$. Thus, the Fourier transform becomes a representation of the signal in the frequency domain. This is illustrated in figure 2.1.

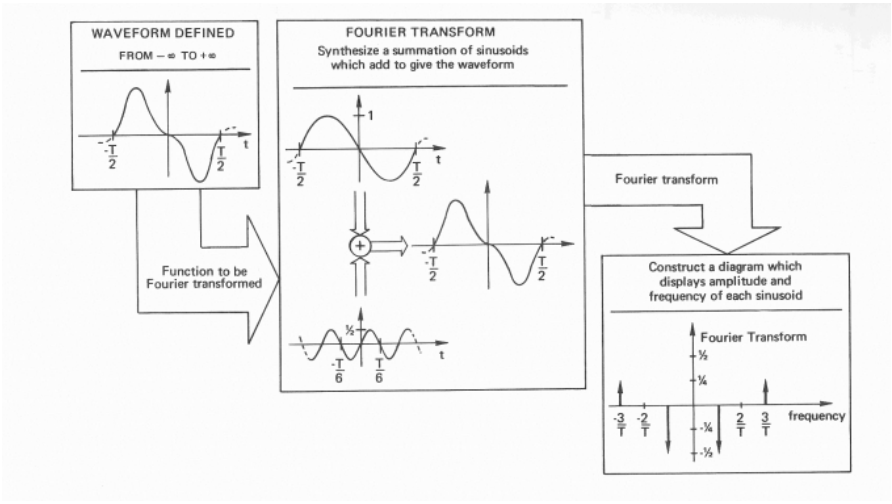


Figure 2.1 – Interpretation of the Fourier transform [2.5]

As can be seen, this process transforms a continuous function into a discrete function. If the input were sampled, the inverse would be true, creating a continuous Fourier series function. In order to be computed effectively on digital computers (which cannot process unsampled continuous functions), the discrete Fourier transform was developed to modify data samples to be suitable for computation [2.6]. Through sampling and windowing a continuous input sample, the relationship in equation 2.2 is established [2.5].

$$\overline{S}\left(\frac{n}{NT}\right) = \sum_{k=0}^{N-1} h(kT) e^{\frac{-j2\pi k}{N}} \qquad n = 0,1,\ldots,N-1 \tag{2.2}$$

for N samples over a windowing period of T. This function is a digital approximation of $S(f)$, with both time and frequency domains given as discrete series. However, this is computationally very intensive, requiring N^2 complex operations. As data spaces increase in size, this number escalates quickly. By factorising the calculation in matrix space, the number of operations is reduced to $N\log N$ [2.7], which yields a major improvement for large datasets.

The FFT has become an established tool in digital signal and data processing, and finds widespread use in nearly all technical fields.

2.2.2 Wavelet Processing and Classification

Wavelets build on the principles of Fourier transforms – to represent a complex signal by a series of simpler signals. Where Fourier utilises continuous sinusoids to build up a signal, wavelet processing decomposes a signal into a series of signals that share basic uniform properties (or bases) [2.8]. These wavelets have the advantage of being able to scale the resolution of each wavelet, allowing a signal to be built up efficiently using multiple resolutions of the same wavelet bases. This allows information density to be local and changing, as opposed to Fourier's uniform information density [2.9].

The simplest wavelet base is Haar's wavelet [2.6] [2.10], whose wavelet function $\psi(x)$ is defined as

$$\psi(x) = \phi(2x) - \phi(2x-1) \tag{2.3}$$

where *φ(x)* is a recursive scaling function

$$\phi(x) = \sum_n h(n)\sqrt{2}\phi(2t-n) \tag{2.4}$$

where h(n) is a sequence of constant coefficient terms. In the case of Haar's wavelet, this scaling function is defined as

$$\phi(x) = \phi(2x) + \phi(2x-1) \tag{2.5}$$

In figure 2.2, the Haar wavelet has been applied to a Doppler function. The wavelet decomposes the signal by the rate of change for a given resolution period. Small resolutions highlight periods of rapid change, and large resolutions highlight periods of slow change.

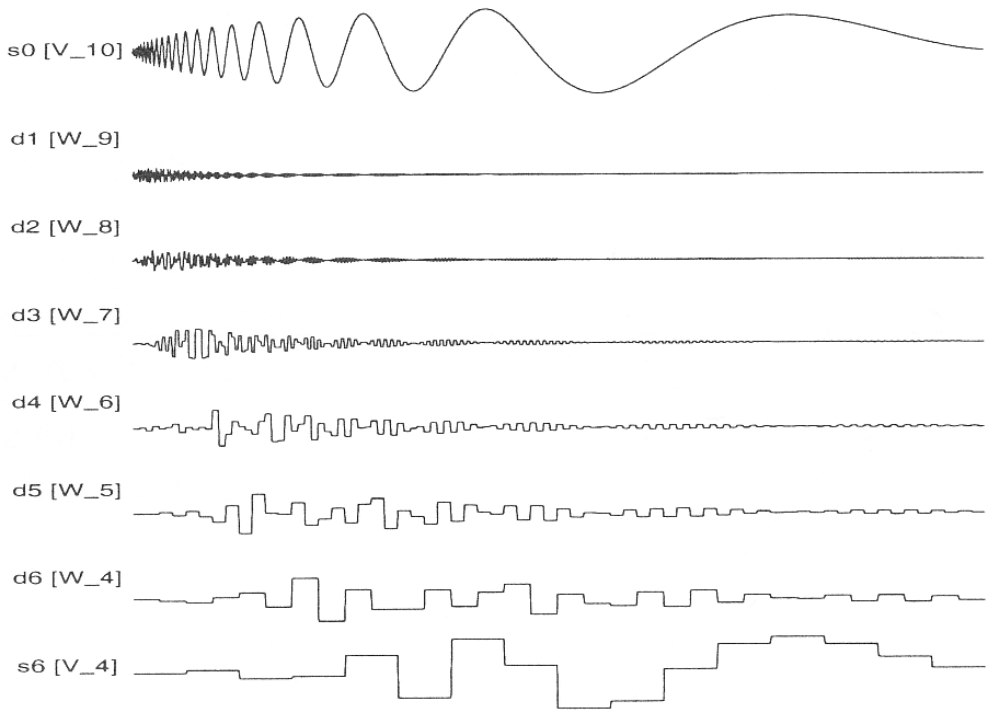


Figure 2.2 Haar decomposition of a Doppler function [2.10]

Other, more complex, wavelets have been developed, such as Shannon's wavelets [2.11] and Daubechies's wavelets [2.12] for use in practical fields ranging from compression, filtering and signal analysis.

2.3 Application of Time-dependent Techniques in Chemical Sensing

The time-dependent methods presented have been used in many technical fields, from signal analysis, data mining, compression, filtering and other computational fields. However, with the advent of advanced artificial olfactory devices, the need for these more advanced processing techniques has arisen. This section explores how these methods are currently being applied in the field of chemical sensing.

2.3.1 Fast-Fourier Transform

The FFT process has been applied to quantification problems, such as thermal cycling [2.13] and classification problems, such as distinguishing between petroleum vapour and CO [2.14]. Such examples show the FFT to be useful for quantification purposes, utilising the magnitude of selected harmonics to differentiate between concentrations or environments. However, classification is challenging, as the harmonics for similar analytes fall in similar regions, making dissemination between them near impossible. As such, it can be seen that the

FFT can form a good basis for the development of quantification techniques, but lack the ability to classify data, and is thus not a method that can be universally applied to many problems.

2.3.2 Wavelet Processing and Classification

Wavelet processing entered the field of chemical sensing as a method for processing electronic tongue data [2.15], a device which detects chemicals in liquid state, primarily consisting of SAW or QMB sensors. The method was then developed further, adapted to gaseous QMB sensors [2.16] and MOX sensors [2.17]. These applications are in the realm of classification problems, and demonstrate how the limitations of the FFT are overcome when using wavelet decompositions. Of particular note is the way in which wavelets process and analyse data in both the frequency and time domains, as opposed to the FFT which only exists in the frequency domain. By analysing the data at a select combination of resolutions, the classification limitation of the FFT is overcome by wavelet processing.

2.4 Advantages and Disadvantages of using Time-dependent Techniques

Time-dependent processing techniques allow the appreciation of data in another domain, in turn making time an important variable for the study of information.

This opens up new information already contained within data for processing, leading to the more effective use of that information and the solving of more complex problems. However, such methods are very computationally complex, often requiring many stages to prepare the data before processing. Algorithms such as the FFT help alleviate this problem, but still remain computationally intensive when compared to time-indepenent processing approaches.

In the next chapter another time-dependent technique is examined, which so far has had no application in the field of chemical sensing.

2.5 References

[2.1] M. M. Mozell, "Evidence for a Chromatographic Model of Olfaction", *Journal of General Physiology*, vol. 56, pp 46-63, 1970

[2.2] D. F. Proctor and I. Andersen, *The Nose: Upper Airway Physiology and the Atmospheric Environment*, Amsterdam; New York: Elsevier Biomedical Press, 1982

[2.3] P. Vroon, *Smell: The Secret Seducer*, New York: Farrar Straus Giroux, 1997

[2.4] H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*, Berlin; New York: Springer-Verlag, 1981

[2.5] E. O. Brigham, *The Fast Fourier Transform and its Applications*, Englewood Cliffs, N.J.: Prentice Hall, 1988

[2.6] C. S. Burrus *et al*., *Introduction to Wavelets and Wavelet Transforms*, Upper Saddle River, N.J.: Prentice Hall, 1998

[2.7] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Amsterdam; London: Elsevier, 2009

[2.8] F. Keinert, *Wavelets and Multiwavelets*, Boca Raton; London: Chapman & Hall/ CRC Press, 2004

[2.9] P. J. Van Fleet, *Discrete Wavelet Transformations*, Hoboken, N.J.: Wiley-Interscience, 2008

[2.10] B. Vidakovic, *Statistical Modelling by Wavelets*, New York; Chichester: John Wiley & Sons, 1999

[2.11] M. Holschneider, *Wavelets: An Analysis Tool*, Oxford: Clarendon Press, 1995

[2.12] Y. Y. Tang, L. H. Yang, J. Liu and H. Ma, *Wavelet Theory and Its Application to Pattern Recognition*, Singapore; River Edge, N.J.: World Scientific, 2000

[2.13] W. M. Sears, K. Colbow, R. Slamka and F. Consadori, "Selective Thermally Cycled Gas Sensing Using Fast Fourier-Transform Techniques", *Sensors & Actuators B*, vol. 2, pp. 283-289, 1990

[2.14] X-J. Haung, Y-K. Choi, K-S. Yun and E. Yoon, "Oscillating behaviour of hazardous gas on tin oxide gas sensor: Fourier and wavelet transform analysis", *Sensors & Actuators B*, vol. 115, pp. 357-364, 2006

[2.15] T. Artursson and M. Holmburg, "Wavelet transform of electronic tongue data", *Sensors & Actuators B*, vol. 87, pp. 379-391, 2002

[2.16] G. J. Gouws and D. J. Gouws, "Analyte identification using concentration modulation and wavelet analysis of QCM sensors", *Sensors & Actuators B*, vol. 91, pp. 326-332, 2003

[2.17] H. Ding, H. Ge and J. Liu, "High performance of gas identification by wavelet transform-based fast feature extraction from temperature modulated semiconductor gas sensors", *Sensors & Actuators B*, vol. 107, pp. 749-755, 2005

Chapter 3: Convolution and its use in Chemical Sensing

3.1 Introduction

In the previous chapter, several methods of processing and identifying features in time-dependant data were presented. These methods are effective when used appropriately, but often computationally complex and processor intensive. In systems where speed and simplicity are desired, this poses a barrier to their use. This chapter introduces how convolution can be utilised to process time-dependant data from chemical sensors; a process that is much less computationally intensive, allowing the process of feature extraction to be performed effectively and quickly in less complex devices.

3.2 The Convolution Function

The convolution function is a method of combining signals. The function integrates the product of one signal, *f(t)*, and another, time-reversed, signal, *g(t)*, over the time domain [3.1]

$$f(\tau) * g(\tau) = \int_{-\infty}^{\infty} f(t)g(\tau - t)dt \tag{3.1}$$

The combined signal peaks where both signals are best matched, producing the highest integration area when multiplied. A graphical representation of this process is show in Figure 3.1.

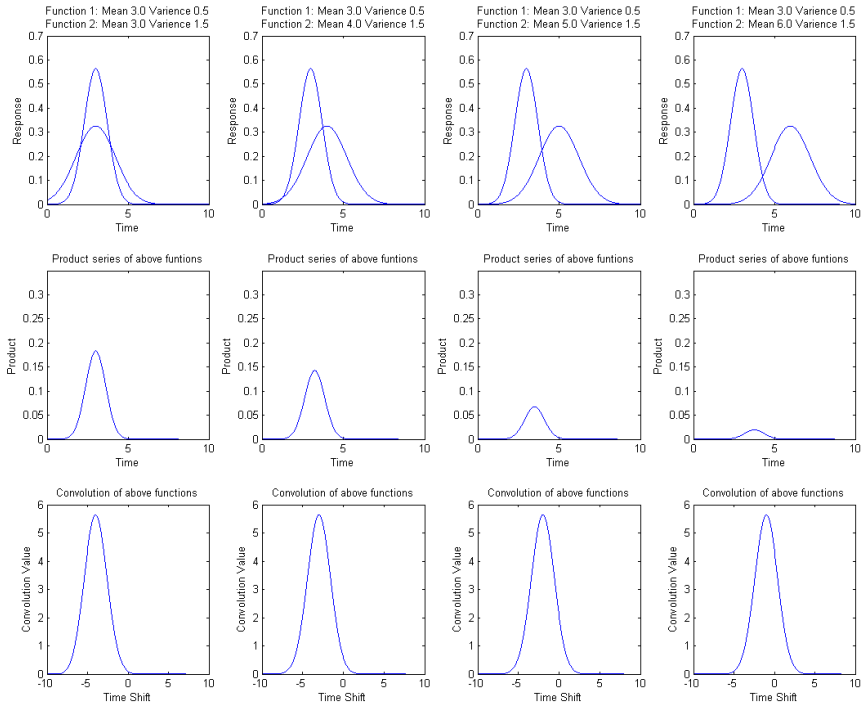


Figure 3.1 Graphical representation of the convolution process between two signals.

The convolution function has several useful properties, foremost of which is its simplicity once transformed to the frequency domain by either a Fourier or Laplace transform [3.1].

$$f(t) * g(t) \xrightarrow{Laplace} F(s)G(s) \tag{3.2}$$

By turning an integral into a multiplication, the convolution function is widely used in system analysis and superposition analysis to define the output of a system, and allows the process of convolution to be computationally efficient. Once this operation has been carried out in the frequency domain, a reverse transform can be performed to retrieve the convolution signal.

3.3 Convolution and Chemical Sensors

Convolution is usually used in system analysis to operate a transfer function on an input signal to define an output signal. However, it can also be used to seek out similarities between two signals [3.2]. The peaks and troughs of the convolution signal highlight periods where the two component signals are most and least similar.

Chemical sensing has long proved to be a difficult challenge for classification systems, particularly in the area of odour recognition [3.3]. Here, an analytical approach does not work, as it is possible for two odours to possess the same chemical and structural layout, but isomerism causes the odour to be perceived by the olfactory system differently. More complex odours can also be altered by the smallest taint or imbalance of the respective molecules that compose it [3.4].

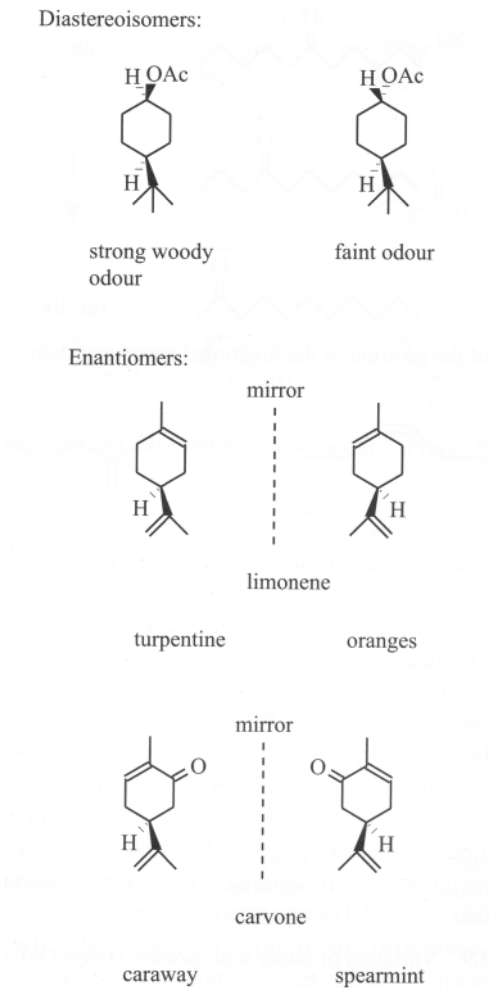


Figure 3.2 Examples of diastereoisomers and enantiomers, whose different structures produce noticeably differing odours. [3.4]

Chemical sensors are used to detect these differences, but due to the limitations presented in the previous chapter, challenge classification systems. The large number of variable conditions to which these sensors are sensitive (temperature, humidity, air, poisoning), while allowing for broad classifications of distinct gases, struggle to identify small changes or taints between very similar gases.

Convolution has been proposed as a method for detecting such taints. Chemical sensor arrays often contain several differing sensing materials with partial

specifity. By using convolution to identify the similarities and differences between these signals, this convolution or characteristic signal for the input analyte can be compared to other analytes, with any differences between the characteristic signals betraying differences or taints.

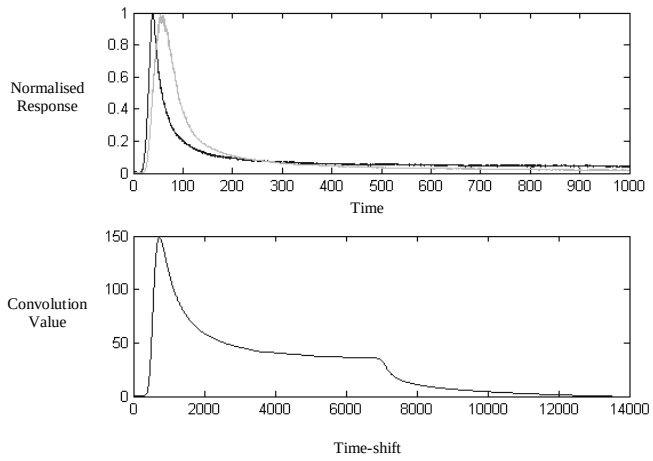


Figure 3.3 Example of a convolution characteristic signal

3.4 Using Characteristic Signals

This suggestion of combining differing signals into characteristic signals and performing comparison is itself a novel idea in the field of chemical and odour recognition. Convolution is not the only method by which sensor signals could be combined to produce characteristic signals. The product of the signals, or the difference between them, could also produce characteristic signals that characterise the analyte.

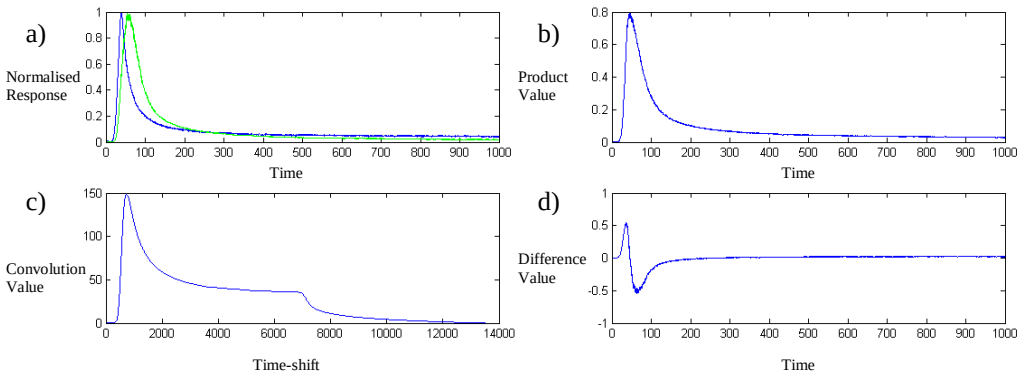


Figure 3.4 Types of characteristic signals (a) original signal (b) product characteristic signal (c) convolution characteristic signal (d) difference characteristic signal

By analysing and comparing properties of these characteristic signals, such an approach can be used in the classification processed for an odour identification problem. By using a pair of signals that differ spatially, by being in a different sensor space or having a different experience, the characteristic signal can be more information rich than either source signal, containing comparisons in and between both the spatial and temporal domain.

Extracting features from this information rich signal, the features can be used in one of the many analysis systems currently in use in the field of chemical sensing, such as linear regression of principal components analysis data or linear discriminant function data. This would make sure the method is easy to adopt into existing systems, simply forming part of the processing chain.

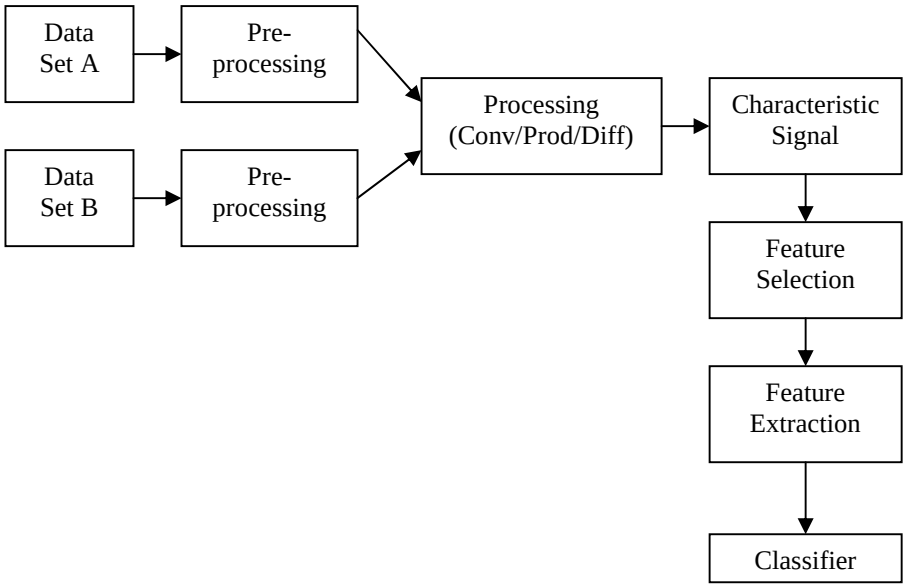


Figure 3.5 A complete processing chain for a system utilising characteristic signals

The simplified processing chain shown in figure 3.5 takes two data sets, A and B, and performs individual pre-processing upon them to prepare the data for processing (such as filtering or normalisation). The data are then processed together, and through convolution (or other signal combination method) a characteristic signal is generated. This characteristic signal is then passed into a feature selection and extraction process, which identify and isolate features suitable for use in the chosen classifier. This feature set is then used to classify the data, using methods such as regression or grouping to define classes.

3.5 References

[3.1] I. I. Hirschman and D. V. Widder, *The Convolution Transform*, Princeton: Princeton University Press, 1955

[3.2] R. D. Hippenstiel, *Detection Theory: Applications and Digital Signal Processing*, Boca Raton; London: CRC Press, 2002

[3.3] T. C. Pierce, S. S. Schiffman, H. T. Nagel and J. W. Gardner, *Handbook of Machine Olfaction: Electronic Nose Technology*, Weinheim: Wiley-VCH, 2003

[3.4] R. R. Calkin and J. S. Jellinek, *Perfumery: Practice and Principle*, New York; John Wiley & Sons, 1994

Chapter 4: Parametric Performance of the Convolution-Based Method on Simulated Chemical Sensor Data

4.1 Introduction

This chapter outlines the process of applying the convolution approach outlined in chapter 3 to a simulated electronic nose system. Alongside this, other signal combination approaches to signal processing are also considered, as well as traditional approaches to processing electronic nose data. Demonstrated is the validation of the techniques presented to classify electronic nose data successfully, and an improvement over current methodologies.

4.2 Simulating E-Nose Data

In order to demonstrate the validity of signal combination techniques, an available simulation is used to provide a proof of concept test bed to explore the use of convolution in signal processing, as well as other similar approaches. The simulation data are processed using current processing methods as a point of comparison.

4.2.1 Outline of E-Nose Simulation

In previous work, an advanced e-mucosa type electronic device, referred to as Protonose III, has been simulated [4.1]. This device consists of four arrays of ten differing partially selective sensors, each array separated along a retentive flow channel, the physical realisation of which is illustrated in Figure 4.1. This type of device is called an artificial olfactory mucosa, or e-mucosa.

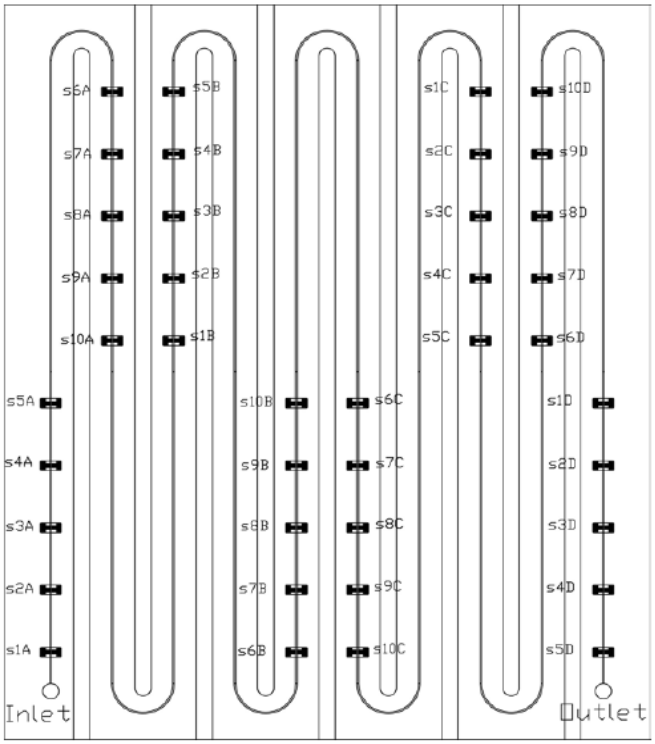


Figure 4.1 Schematic of the physical sensor layout of the Protonose III [4.1]

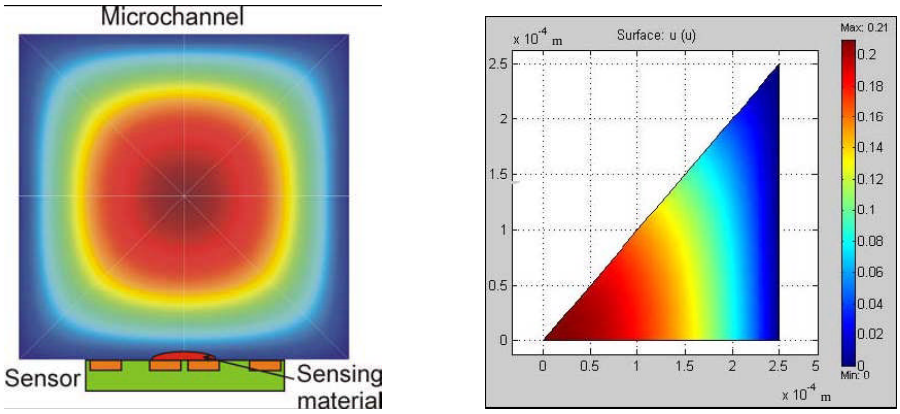
4.2.2 Generation of Simulation Data

The data used in the simulation of tandem e-nose processing are based on the simulation of a gas chromatography microchannel by Tan at the University of Warwick [4.1]. This work generated a basic data set as a proof of concept, which modelled the flow of an analyte gas pulse as it travels down a coated microchannel.

This concept simulation can be used as a basis for generating many data sets by systematically adding noise to extend the base data. As noise models rely on the generation of random numbers and sequences, many permutations of simulated noisy data can be generated, under the control of a small number of parameters.

Presented here is a summary of the simulation carried out by Tan [4.1]. For more detail on the background of this model, please refer to the thesis.

The simulation was carried out using the finite-element simulation package FEMLAB 2.3b. A geometric model of the microchannel was first established – a square channel of dimension 0.5 mm by 0.5 mm. Computational complexity was reduced by only simulating 1/8 of the channel's cross-section, allowed by the symmetry of the simple channel (Figure 4.2).



(a) (b)

Figure 4.2 2-D velocity profile of the mircochannel. (a) Complete channel with sensor placement. (b) 1/8 computational model of the velocity profile. [4.1]

The sensor positions were placed along the length of the channel, although not at the precise locations (Figure 4.3) due to scale limitations imposed by the simulation package. However, as a linear flow model was assumed, scaling of simulation parameters was performed to compensate for this geometric error.

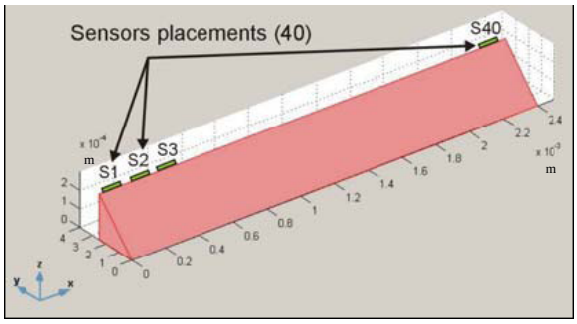


Figure 4.3 Scaled geometric model of the protonose simulation. [4.1]

A retentive polymer coating was present on three sides of the microchannel. Flow of analyte (mobile phase) was then simulated using a single plane of symmetry, governed by a multi-physics system of Navier-Stokes fluid flow (Equation 4.1) and convection-diffusion (Equation 4.2), defined by

$$\mu\nabla^2 V + \rho(V.\nabla)V + \nabla P = 0 \quad (4.1)$$

where μ is the viscosity, ρ is the density, P is the pressure and V is the velocity, and

$$\frac{\partial C}{\partial t} + \nabla.(-D\nabla C + CV) = 0 \quad (4.2)$$

where D is the diffusion coefficient and C is the concentration.

For these flow equations, boundary conditions were established between the microchannel (μC) and the inlet, outlet and symmetry boundaries as follows:

$$C = C_i \ (t < t_{PW}) \quad \text{at} \quad \partial\,\Omega_{\mu C,inlet} \quad (4.3)$$

$$(-D\nabla C + CV).n = (CV).n \quad \text{at} \quad \partial\,\Omega_{\mu C,out} \quad (4.4)$$

$$(-D\nabla C + CV).n = 0 \quad \text{at} \quad \partial\,\Omega_{\mu C,symmetry} \quad (4.5)$$

where C_i is the injected concentration over the period of t_{PW}, and n is the normal vector to the respective boundary.

At the boundary of the retentive polymer coating (stationary phase, SP), a further diffusion model was required to represent the transport of molecules into the coating, defined by

$$\frac{\partial C}{\partial t} + \nabla.(-D\nabla C) = 0 \qquad (4.6)$$

Boundary conditions for the diffusion equations were defined as follows:

$$\left[\frac{\partial C_M}{\partial t} + \nabla.(-D\nabla C_M + C_M V)\right].n = M.(C_S - cC_M) \quad \text{at } \partial\,\Omega_{\mu C,SP} \qquad (4.7)$$

$$\left[\frac{\partial C_S}{\partial t} + \nabla.(-D\nabla C_S + C_S V)\right].n = M.(cC_M - C_S) \quad \text{at } \partial\,\Omega_{SP,\mu C} \qquad (4.8)$$

where n is the normal vector to the respective boundary, and c is the partition coefficient, represented by the ratio of the concentration in the stationary phase, C_S, and the concentration in the mobile phase, C_M.

A separate simulation of the sensor response showed that their response was approximate to a first-order exponential curve (Figure 4.4). Rather than incorporate the sensors directly into the geometric model, introducing further complexity for computation, the analyte profile at each sensor position was used as the input to a first-order response model.

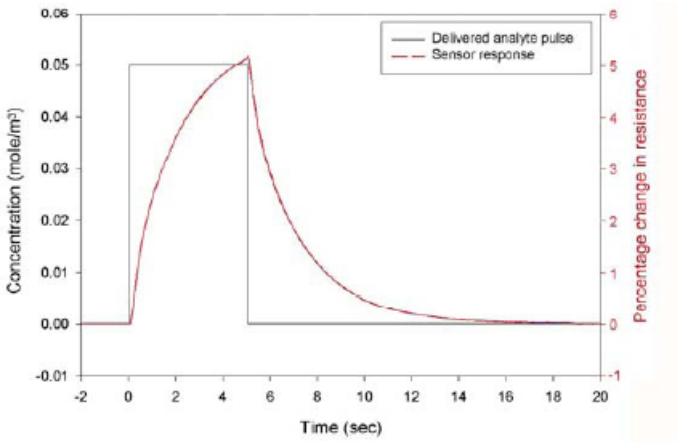


Figure 4.4 Simulation of sensor response (first-order exponential) [4.1]

Using these models, simulations were carried out using the parameters representing three test analytes (ethanol, toluene, 50/50 mixture of ethanol and toluene). The timestep used during the ethanol simulation was 0.2 seconds, and ran for 200 seconds. The toluene and mixture simulations ran with a 0.2 second timestep for 140 seconds, then a 1 second timestep for the next 460 seconds, for a total simulation time of 600 seconds.

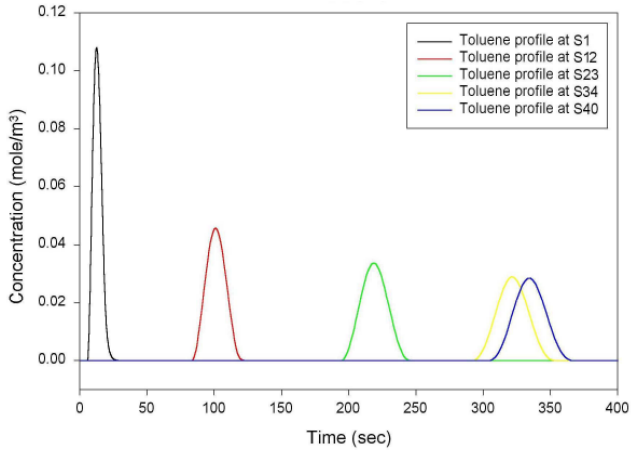


Figure 4.5 Simulation response to toluene at five different sensor positions in the microchannel [4.1]

Figure 4.5 shows the broadening of the peak as the pulse travels along the channel. As no leakage is assumed in this ideal model, the area of each response remains constant.

Within this simulated sensor structure, it is possible to simulate two contemporary electronic nose devices; a basic electronic nose [4.2] and a sensor array preceded by a gas chromatography column, or c-nose [4.3]. The way in which these devices are mimicked by the Protonose simulation is illustrated in Figure 4.6.

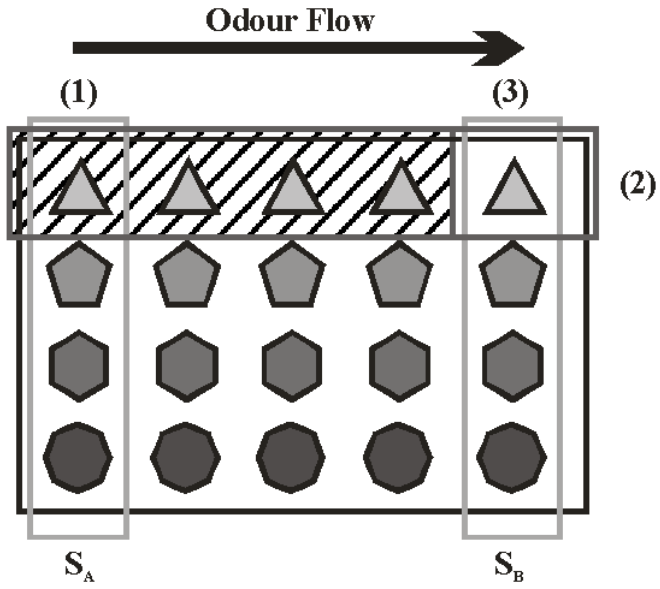


Figure 4.6 Traditional devices contained within the Protonose III e-mucosa. (1) A traditional e-nose. (2) A gas chromatography column (no sensors). (3) A gas chromatography column with an e-nose at the end, or c-nose.

In order to be used, the simulation requires the addition of noise to provide a suitable testing dataset to be used in a classification problem.

4.2.3 Noise Simulation

When a chemical sensing device collects data, it is subject to several sources of noise [4.4]. This creates variations in the data collected, and necessitates signal processing and classification processes in order to reduce noise and extract information. These sources of noise need to be simulated and added to the e-mucosa simulated data set in order to demonstrate the ability of a processing approach to cope with real-world applications, and correctly classify data sets under harsh conditions (e.g. sensor drift and random effects).

Three main categories of noise have been identified, grouped by the manner in which they act on the data.

4.2.3.1 Additive Noise

Additive noise covers noise sources that have an additive effect on the data. This primarily includes most external sources of random noise, but also includes sensor noise and environmental taints. For this simulation, this noise category was modelled as white Gaussian noise, with a mean of 0 and a variance based on a percentage of the sensor's peak magnitude,

$$X_{SNR}(t) = X(t) + n(t) \tag{4.9}$$

where $X_{SNR}(t)$ is the final noisy signal, $X(t)$ is the simulated sensor signal and $n(t)$ is the noise signal, defined by,

$$n(t) = n \times \max(X) \times randn(t) \tag{4.10}$$

where n is the noise percentage (a constant), and *randn* is a random number generated by the MATLAB 7.4.0 (R2007a) *randn* random number generation function, with a mean of 0 and variance of 1.

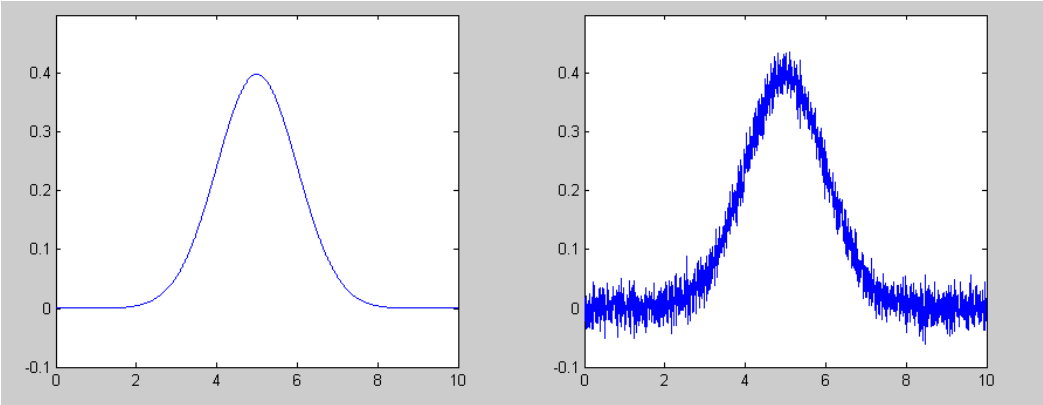


Figure 4.7 A non-scalar example of the effect of additive noise. (a) Source signal – normal curve, mean 5, variance 1. (b) SNR noise with a level of 5% added to source signal.

4.2.3.2 Sample Variance

Sample variance represents the types of noise that cause variation in the magnitude of the entire source signal, such as variations between samples or the response of similar sensors. This noise category is simulated by using a Gaussian factor, with a mean of 1 and a variance as the noise percentage.

$$X_{SV}(t) = X(t)N \tag{4.11}$$

where $X_{SV}(t)$ is the final noisy signal, $X(t)$ is the simulated sensor signal and N is the noise value, defined by,

$$N = 1 + (n \times randn) \tag{4.12}$$

where n is the noise percentage, and *randn* is a random number generated by the MATLAB 7.4.0 (R2007a) *randn* random number generation function, with a mean of 0 and variance of 1.

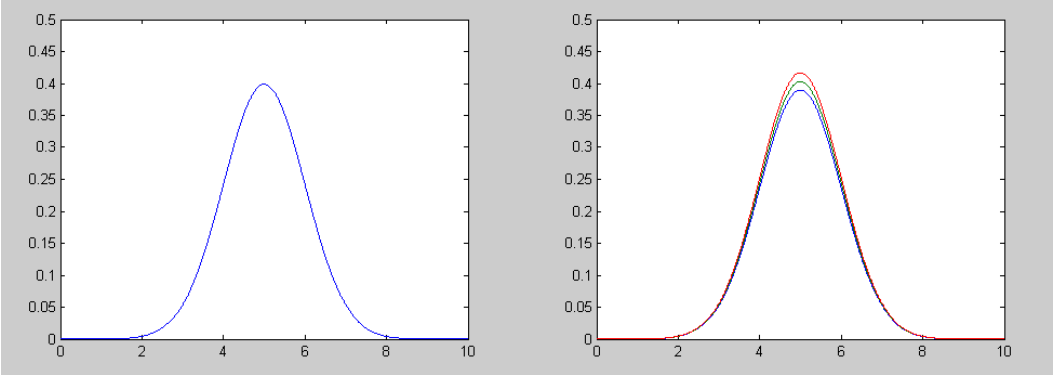


Figure 4.8 A non-scalar example of the effect caused by sample variance noise. (a) Source signal – normal curve, mean 5, variance 1. (b) 3 samples with different magnitudes due to a sample variance noise level of 5%.

4.2.3.3 Baseline Drift

Baseline drift is one of the more problematic noise types facing electronic nose systems [4.5]. The main sources of drift in these systems include aging, poisoning, ambient humidity and temperature change. This noise source is modelled with a Gaussian distribution with mean of 0 and a variance based on a percentage of the sensor's peak magnitude, and added as a constant to an entire sensor response. This models a long term systematic drift of the sensor device due to aging or poisoning and environmental differences between sample collection sessions, not a short-term drift over a single data sample, and assumes that the resulting drift is steady over the period of a single sample.

$$X_{BD}(t) = X(t) + N \tag{4.13}$$

where $X_{BD}(t)$ is the final noisy signal, $X(t)$ is the simulated sensor signal and N the noise value, defined by,

$$N = n \times \max(X) \times randn \tag{4.14}$$

where n is the noise percentage, and *randn* is a random number generated by the MATLAB 7.4.0 (R2007a) *randn* random number generation function, with a mean of 0 and variance of 1.

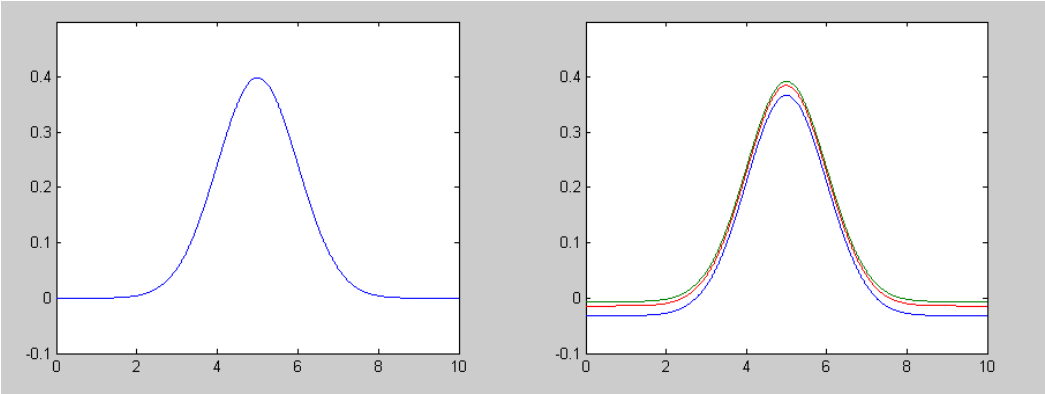


Figure 4.9 A non-scalar example of the effect caused by baseline drift. (a) Source signal – normal curve, mean 5, variance 1. (b) 3 samples with different baselines due to a drift noise level of 5%.

4.2.4 The Final Simulation Data Set

The final data set produced of the simulation is based on the simulations generated from the Protonose III device [4.1], using the commercial software MATLAB 6.0.0.88 (r12) and a finite element model, FEMLAB 2.3b [4.6]. This simulation generated 3 data sets, simulating a 5 second pulse of each of ethanol, toluene and a 50/50 mixture of the two passing through the system. The retentive micro-channel simulated has a cross-sectional geometry of 0.5 mm × 0.5 mm and a length of 2.3 m, and coated with parylene C as the stationary phase. 40 sensors, in 4 groups of 10 differing tunings, are arranged along the micro-channel as shown in Figure 4.1. For this simulation, only the first 10 sensors and the final 10 sensors are required. Figure 4.10 illustrates some examples of sensor outputs from this simulation.

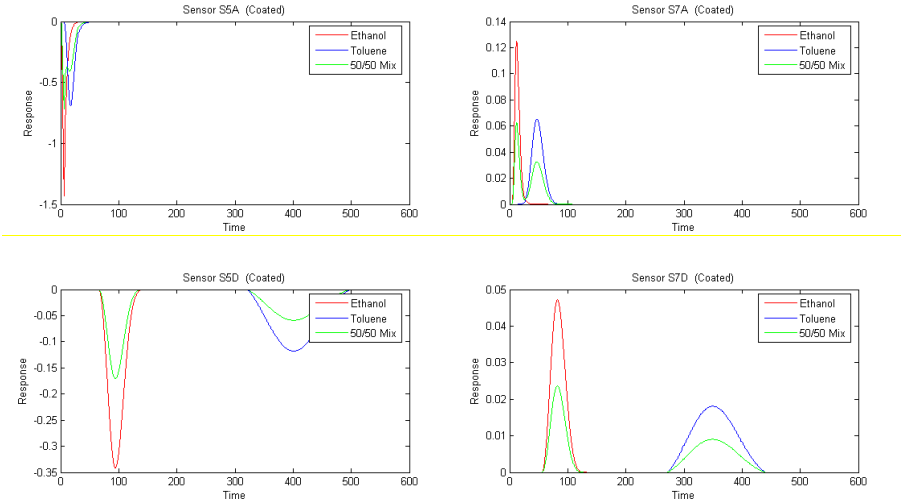


Figure 4.10 Example of 2 paired sensor outputs (sensors 5 and 7) from the simulated Protonose III device. Top – e-nose array. Bottom – z-nose array [4.1]

From each of these 3 data sets, 20 random data sets were generated by the addition of a noise signal category, producing a complete data set of 60 samples containing 3 analytes. This produces 3, 60-sample data sets, one for each of the noise categories; additive noise, sample variance and baseline drift. A further 3 data sets were generated using binary combinations of noise categories; additive and sample variance, additive and baseline drift and sample variance and baseline drift. For all of these data sets, the noise value was set to a 5% level. This is typical for sensor noise (additive and sample variance), but a very high level for drift noise over a typical measurement period.

The result of this data generation results in 6 noisy e-nose data sets and 6 noisy z-nose data sets, totalling 240 data samples. This broad range of data sets will allow a detailed analysis of the performance of each processing method with respect to the major noise types that can influence a chemical sensing result.

4.3 Processing the Simulation Data

With the simulated data sets complete, a processing strategy must be decided upon. This will comprise of a pre-processing stage and a processing stage. This prepares the data for use in a classification scheme. The purpose of the pre-processing is to prepare the data for processing, ensuring that data are weighted correctly and eliminating any potential biasing factors for the data. Following this, the data can be processed, selecting and extracting information and data that will be most advantageous for use in the classification stage, and provide the best discrimination between classes. All the following processing was performed using the commercial software MATLAB 7.4.0 (R2007b), and the implementation of these functions can be found in Appendix A.

4.3.1 Pre-Processing

The output from real sensors can vary significantly in magnitude, while the information contained within each is equally important. Normalisation ensures that such variable data values are all treated in an unbiased manner. However, any noise on small signals will be amplified greatly, and so such operations must be used with careful consideration. Two normalisation approaches were considered; auto-scaling and auto-ranging.

Auto-scaling linearly scales the magnitude of all data points in a given sample so that their values lie in the range between 0 and 1, and is signed to preserve the direction of the magnitude. Thus, the calculation for $S^j(t)$, the pre-processed sensor signal

$$S^j(t) = \frac{\left|X^j(t)\right|}{\max\left(\left|X^{\forall j}\right|\right)} \tag{4.14}$$

where $X^j(t)$ is the noise simulation signal, is performed. In removing the length of the vector, balancing it at 1 for all vectors, concentration is removed from the signal information, which makes this function more suited for establishing qualitative information (i.e. classification problems) rather than quantitative information, assuming the data is linear.

Auto-ranging linear scales the magnitude of the spread of the sensor data so that the range is equal to 1. This method will preserve any information contained within small signals in the data, at the disadvantage of biasing signals with large values. More complex problems may have sensors that respond in both directions, information that would be lost in regular auto-scaling. The auto-ranging technique is derived to overcome this problem. However, the same removal of concentration information occurs, making this method most suitable for classification problems and not quantification problems. Thus, auto-ranging is performed to produce a pre-processed sensor signal

$$S^j(t) = \frac{\left|X^j(t)\right|}{\max\left(\left|X^{\forall j}\right|\right) - \min\left(\left|X^{\forall j}\right|\right)} \tag{4.15}$$

To gauge the advantages of pre-processing, the raw data, with no pre-processing, will also be processed.

$$S^j(t) = X^j(t) \tag{4.16}$$

All these pre-processing approaches are statistical in nature, intended for use with statistical processing and classification techniques, and assume linearity in the initial sensor signals.

4.3.2 Processing

Having prepared the data sets, processing of the data can be done. The main requirement of the central processing stage is to develop features so that they can be extracted for use in a classification scheme. This feature is chosen to be a representative distinguishing factor that can be used to differentiate between samples and ultimately used to identify them.

4.3.2.1 Peak Magnitude

Peak magnitude is a feature whose use is prevalent in chemical sensing [4.2] [4.7] [4.8]. As such, it represents a baseline from which any improvements can be judged. The pre-processed data sets for both sets of sensors, the front (S_A, e-

nose analogue) and the back (S_B, c-nose analogue), are processed in this manner, producing an array of peak magnitude values suitable for use in classification.

4.3.2.2 Cross-Convolution

Cross-convolution incorporates a convolution characteristic signal, from which a feature can be defined. Pairs of sensors within the array have convolution performed on them, producing a group of characteristic signals. Again, this process was performed on both the front sensors (eq. 4.17) and the back sensors (eq. 4.18).

$$y_A{}^{ij}(t) = \int_{\tau=-\infty}^{\infty} S_A^i(\tau) S_A^j(t-\tau) d\tau \tag{4.17}$$

$$y_B{}^{ij}(t) = \int_{\tau=-\infty}^{\infty} S_B^i(\tau) S_B^j(t-\tau) d\tau \tag{4.18}$$

As there are 10 sensors, 10 sensor pairings are chosen, so the dimension of the data is comparable to the peak magnitude feature set, and using all 45 possible convolution pairings would add excessive redundancy and magnitude any noise or errors still present in the signals. These pairings are illustrated in Figure 4.10.

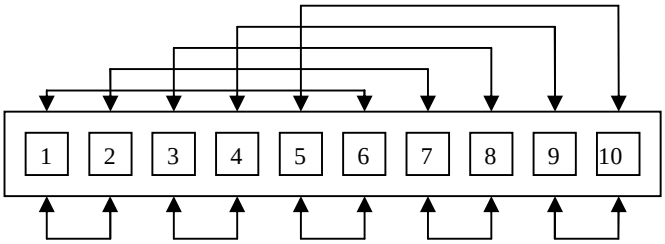


Figure 4.11 Selection of sensor pairings for cross-convolution calculations

Referring to Figure 4.1, the pairings illustrated above the sensors represent sensors which are equidistant within a group of sensors in the array, while the lower sensor pairings are selected based on order.

From this set of characteristic signals, the area under the curve was computed as the feature for use in classification. This feature was chosen as it contains information on the overall shape of the signal, which is how the characteristic signal varies with differences in the source samples.

4.4 Post-Processing and Classification

The processing results in a total of 72 feature sets (6 noise types × 2 devices × 3 pre-processing methods × 2 processing methods). These features are used in a classification scheme. Firstly, a principal components analysis is applied. This produces a set of principal component sets that can produce a linear graphical output in 2 or 3 dimensions. Upon these principal component data sets, linear regression would normally be used to produce classification regions. However

here, a t-test will be performed to generate a quantitative measure of the quality of each of the feature sets.

4.4.1 Principal Components Analysis

Principal components analysis (PCA) is a method where data which contain many dimensions are converted in such a manner that the majority of the data's variance is contained in a low number of dimensions [4.9]. The PCA was performed using the commercial software MATLAB 7.4.0 (R2007a), and the PCA function contained within its statistical toolbox. The results of the PCA showed that in all cases, 95%+ of the variance was contained within the first 2 principal components. The PCA data sets were plotted using the first 2 principal components, a selection of which are shown in Figure 4.11 (the full selection of plots can be found in Appendix B).

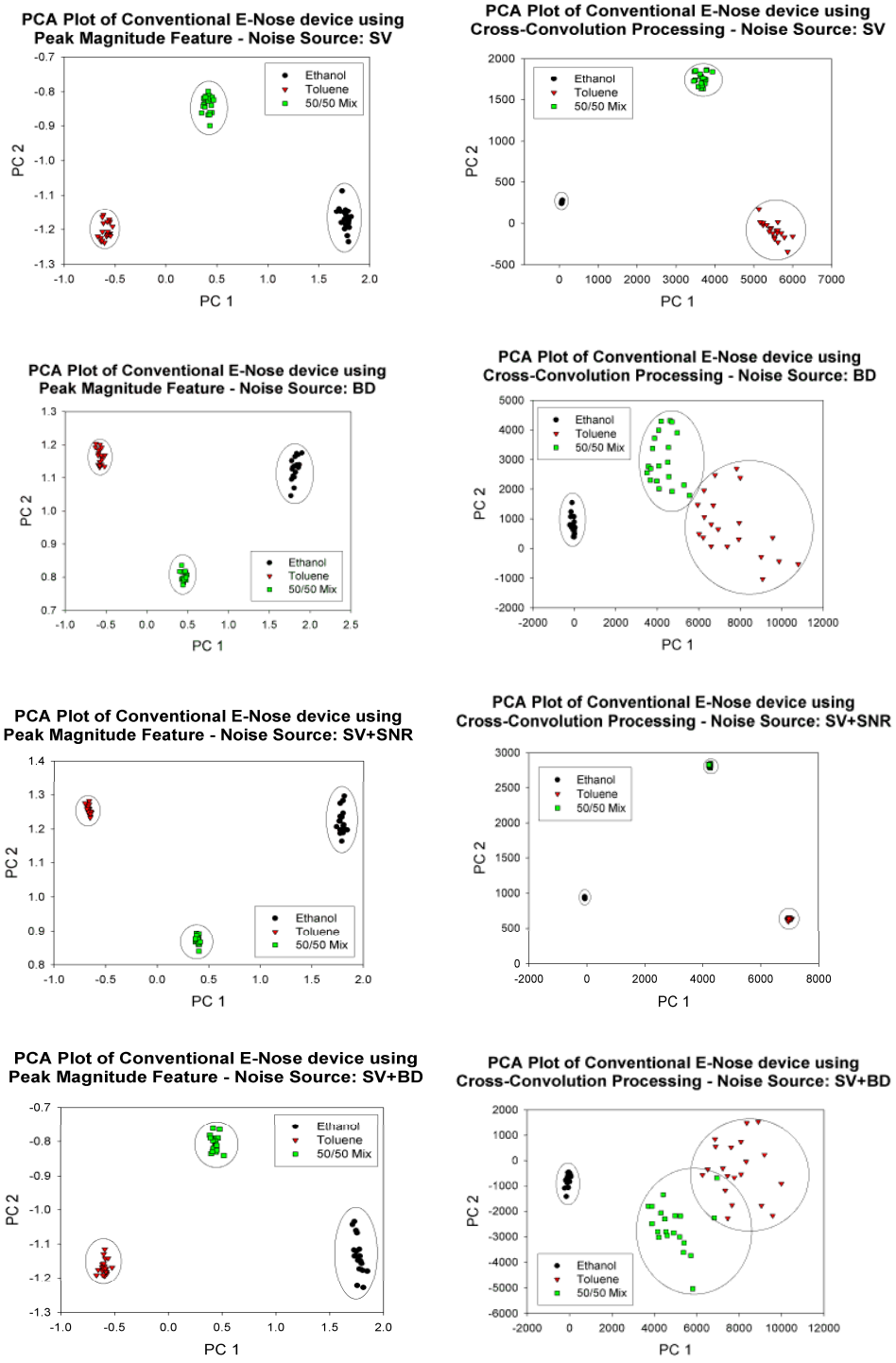


Figure 4.12 Selection of PCA plots after processing. Left – traditional electronic nose. Right – cross-convolution on electronic nose data. Top to bottom – sample variance noise, baseline drift, SV + SNR, SV + BD.

4.4.2 Paired t-test

A PCA test forms a good foundation for classification using regression techniques, but does not provide a good quantitative measure for use in assessing the quality of one technique against another. Using information generated during the PCA process that indicates the percentage of the variance contained within each principal component, a t-test can be performed to generate a quantitative measure of the quality of the PCA. The paired t-test considers the relative size and separation of the data groups, expressed as a t-score. These t-scores were generated between each pair of analytes for the first 2 principal components using a separate variance t-test performed in Unistat 5.5 software. Tables 4.1-4.3 show the t-scores generated by this processing for the complete set of feature sets.

		t-scores							
		E-Nose		C-Nose		X-Convolution		CX-Convolution	
		PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2
SV	Eth-Tol	198.13	3.86	146.08	6.08	139.65	118.73	108.68	156.45
	Eth-Mix	121.97	35.96	98.14	24.77	185.53	14.26	59.64	104.68
	Tol-Mix	87.53	46.31	100.70	54.52	56.57	85.13	106.89	10.35
SNR	Eth-Tol	370.43	3.91	352.92	13.83	659.52	64.93	832.03	53.92
	Eth-Mix	214.59	40.94	239.27	59.75	360.31	237.24	422.28	459.31
	Tol-Mix	226.98	91.91	227.79	103.27	254.54	395.24	419.81	400.44
BD	Eth-Tol	226.20	3.52	172.90	6.20	29.55	7.00	24.72	4.46
	Eth-Mix	141.15	39.58	131.88	27.45	24.68	7.45	7.95	12.80
	Tol-Mix	123.10	59.09	81.63	50.81	4.60	12.77	19.63	9.54
SV+SNR	Eth-Tol	405.74	4.20	400.96	16.18	583.41	94.56	849.18	50.48
	Eth-Mix	243.05	43.91	289.17	70.47	385.62	349.25	404.56	446.85
	Tol-Mix	243.54	96.90	222.83	123.24	213.02	474.22	437.58	401.15
SV+BD	Eth-Tol	231.34	2.39	234.48	7.33	35.84	4.76	22.84	4.43
	Eth-Mix	130.73	25.58	177.18	35.46	20.10	8.25	7.10	12.85
	Tol-Mix	107.21	52.78	90.39	52.51	7.36	12.03	17.99	8.38
SNR+BD	Eth-Tol	237.46	3.05	193.47	6.66	28.02	8.73	22.20	4.53
	Eth-Mix	131.48	28.77	122.83	30.77	28.75	8.91	7.71	11.44
	Tol-Mix	145.18	56.94	107.90	46.75	3.81	16.48	16.49	8.61

Table 4.1 t-scores generated between each pair of analyte groups for all the processed feature sets, using a separated variance t-test using an auto-scaling pre-processing technique

		t-scores							
		E-Nose		C-Nose		X-Convolution		CX-Convolution	
		PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2
SV	Eth-Tol	190.65	5.36	179.78	7.71	140.14	129.59	126.63	183.80
	Eth-Mix	124.48	29.22	114.79	35.01	22.71	18.10	46.55	181.68
	Tol-Mix	122.13	59.62	121.00	49.83	63.83	112.39	117.92	15.67
SNR	Eth-Tol	350.36	11.21	439.10	13.77	424.07	220.66	260.40	255.32
	Eth-Mix	221.48	55.08	280.99	55.93	629.29	2917	163.75	175.28
	Tol-Mix	196.47	93.29	196.76	116.80	224.04	220.83	455.17	18.52
BD	Eth-Tol	191.49	4.34	225.96	7.96	21.09	4.92	13.60	3.62
	Eth-Mix	121.95	23.21	161.95	30.95	15.20	4.74	0.43	7.15
	Tol-Mix	97.39	46.07	97.02	75.88	1.32	9.01	13.07	3.30
SV+SNR	Eth-Tol	336.98	9.15	492.66	16.75	394.94	240.12	382.67	272.67
	Eth-Mix	219.05	47.18	333.96	76.44	469.54	31.95	112.39	122.88
	Tol-Mix	207.78	101.34	238.75	128.46	152.38	236.16	281.10	15.63
SV+BD	Eth-Tol	221.32	6.20	185.00	7.16	18.23	4.36	14.14	5.33
	Eth-Mix	138.47	33.67	123.64	31.16	22.03	4.00	5.26	5.51
	Tol-Mix	117.00	51.24	91.64	46.28	2.26	8.02	17.30	1.99
SNR+BD	Eth-Tol	199.73	6.21	214.46	7.60	16.95	3.06	12.65	6.50
	Eth-Mix	130.69	32.35	140.49	31.03	17.43	5.72	1.58	7.43
	Tol-Mix	96.50	67.68	115.20	48.60	3.48	8.52	16.45	2.47

Table 4.2 t-scores generated between each pair of analyte groups for all the processed feature sets, using a separated variance t-test using an auto-ranging pre-processing technique

		t-scores							
		E-Nose		C-Nose		X-Convolution		CX-Convolution	
		PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2
SV	Eth-Tol	219.63	5.45	187.79	6.25	150.94	135.15	135.94	146.37
	Eth-Mix	128.82	29.20	130.72	25.79	130.28	15.94	35.39	149.07
	Tol-Mix	109.13	50.45	84.41	50.02	43.75	98.13	111.32	23.68
SNR	Eth-Tol	274.61	8.30	459.78	14.80	355.44	206.12	254.70	251.66
	Eth-Mix	171.68	40.57	324.15	72.70	441.77	26.37	111.26	122.82
	Tol-Mix	168.90	83.82	194.35	107.66	158.26	213.45	244.61	5.52
BD	Eth-Tol	242.04	6.46	213.07	7.62	16.32	7.08	13.24	4.65
	Eth-Mix	151.31	36.27	141.55	30.18	15.89	5.45	3.74	6.75
	Tol-Mix	104.15	58.64	113.18	67.07	2.14	9.81	14.56	1.76
SV+SNR	Eth-Tol	318.04	11.79	418.27	13.44	44.08	216.35	351.88	252.10
	Eth-Mix	199.32	65.88	296.34	5921	482.30	29.81	185.67	197.26
	Tol-Mix	196.40	86.71	186.68	109.46	176.17	266.27	435.49	18.03
SV+BD	Eth-Tol	238.86	6.12	206.40	8.18	17.83	5.97	13.61	3.90
	Eth-Mix	158.14	32.62	125.29	31.28	18.64	3.94	1.61	9.01
	Tol-Mix	110.48	57.78	79.75	48.22	1.29	10.45	16.15	3.28
SNR+BD	Eth-Tol	267.40	7.25	156.73	6.09	19.62	5.64	14.04	6.46
	Eth-Mix	163.23	34.66	108.70	24.65	16.55	4.11	1.72	7.10
	Tol-Mix	97.38	42.97	100.47	60.53	0.48	8.14	11.78	2.12

Table 4.3 t-scores generated between each pair of analyte groups for all the processed feature sets, using a separated variance t-test using no pre-processing

4.5 Analysis of Results

Analysis of the principal component plots show that all the devices and methods described above successfully separated the three analytes into good groupings that would easily be identified by utilising linear regression. On visual inspection, it is however difficult to quantify the differences between the various plots, due differing scaling factors and tight, well-spaced groupings. However, the t-scores for the cross-convolution methods show a marked improvement over their traditional counterparts in the presence of sample variance or additive noise sources, but showed a markedly lower performance in the presence of baseline drift noise, producing the lowest t-scores and least grouped plots whenever baseline drift was present. The e-nose and c-nose performed similarly in both processing categories. These results also show that auto-scaling is the preferable pre-processing approach.

4.6 Conclusions

In this chapter, an implementation of a convolution processing methodology for use with chemical sensors has been successfully created. The basic simulation data sets have been taken and noise added to create a meaningful test bed to demonstrate the implementation of the convolution methodology alongside a traditional approach based on methods currently in use in chemical sensing. The full processing process has been mapped out, with attention paid to finding the

optimum combination of processing techniques to make the most of this novel approach. Finally, a classification scheme has been suggested and implemented, with an additional quantitative test added so that the various techniques combinations can be assessed.

The results have validated the use of convolution as a processing method for use in classification problems within the field of chemical sensing, demonstrated by the successful separation of the analyte groups. However, the problem solved was not one where the traditional methods failed, but the additional quantitative t-test demonstrates that groupings in the convolution processing method have some superior properties to the traditional approaches, although there exists a definite weakness to baseline drift. As this test did not contain any data with +/- swing, it was not unexpected when the auto-scaling pre-processing method was found to be superior. Based on this success, in the next chapter, the convolution method will be applied to real sensor data, under a similar set of processing conditions.

4.7 References

[4.1] S. L. Tan, “Smart Chemical Sensing: Towards a Nose-on-a-Chip,” Ph.D. dissertation, School of Engineering, University of Warwick, Coventry, UK, 2005

[4.2] J. W. Gardner and P. N. Bartlett, *Electronic Noses: Principles and Applications.* Oxford: Oxford University Press, 1999

[4.3] A. K. Wanekaya *et al.*, "Multicomponent analysis of alcohol vapors using integrated gas chromatography with sensor arrays", *Sensors and Actuators B*, vol. 110, pp. 41-48, 2005

[4.4] C. Distante *et al.*, "On the study of feature extraction methods for an electronic nose", *Sensors and Actuators B*, vol. 87, pp. 274-288, 2002

[4.5] P. Mielle *et al.*, "Electronic Noses: Specify or Disappear", *Sensors and Actuators B*, vol. 69, pp. 287-294, 2000

[4.6] S. L. Tan, J. A. Covington, J. W. Gardner, T. C. Pearce, "Finite element simulation of a biomimetic olfactory microsystem for spatio-temporal signal generation", *AsiaSim 2007*, Springer, 2007, pp. 216-226

[4.7] M. A. Ryan *et al.*, "Monitoring Space Shuttle Air Quality Using the Jet Propulsion Laboratory Electronic Nose", *IEEE Sensors*, vol. 4, no. 3, pp. 337-347, June 2004

[4.8] R. C. Young, W. J. Buttner, B. R. Linnell, R. Ramesham, "Electronic nose for space program applications", *Sensors and Actuators B*, vol. 93, pp. 7-16, 2003

[4.9] A. A. Afifi and V. Clark, *Computer-Aided Multivariate Analysis 3rd edition*, London: Chapman & Hall, 1996

Chapter 5: Parametric Performance of the Convolution-based Method on Real Chemical Sesnor Data: Classification

5.1 Introduction

As presented in the previous chapter, the convolution processing method proved successful for use in classifying simulated data from e-nose and e-mucosa devices, and showed evidence of being superior to traditional methods in certain cases. However, the time-series simulation was an ideal scenario with linear fluid flow and rectilinear geometry. For a more thorough examination of the value of this methodology, the process will now be applied to experimental data gathered from real sensors and their response to simple and complex odours.

5.2 Experimental Data Set

First a data set kindly supplied by ISOCS [5.1] has been used for testing the convolution processing method on real sensor data. This data set describes a set of data collected over period of 3 months, including four analytes; anisol, cyclohexanol, propanol and toluene. The data were provided in MultiSens XML

format [5.2]. Each data set contains responses from two types of chemical sensors, metal oxide resistive gas sensors (MOX) and quartz micro-balance sensors (QMB) with differing polymer coatings, each with eight tunings.

5.3 Applying the Convolution Processing Method to Real Data

The data were converted from MultiSens XML format into a .CSV data format suitable for use with the MATLAB integration of the convolution processing function, detailed in Appendix A. The processing approach used in the simulation is used as the basis for determining the processing scheme here. Based on this, pre-processing will be an auto-scaling normalisation process (equation 4.14), followed by the convolution processing.

Following processing, feature selection will be expanded to consider other possible features that can be obtained from the characteristic signals. Area, magnitude and combinations of these features will be considered as possible time-independent features. A closer examination of suitable classification schemes will also be carried out by comparing a PCA regression with a probabilistic neural network (PNN).

5.3.1 Pre-Processing and Sensor Selection

Following the results of the simulation, the data undergo auto-scaling normalisation, so that the values of the data are scaled between 0 and 1, removing any concentration bias that may be present in the system, but this process will magnify any sampling or variance noise when the signals are small. By using a convolution processing approach, the effect of these noise sources on the final classification results should be reduced, due to the integration process which convolution includes.

Convolution characteristic signals will be generated between pairs of sensors. With 2 different sensor arrays available, there are several pairing options. As there are 8 sensors in each array, a target of 8 characteristic signals is chosen. These 8 signals can be generated by pairing sensors within either the MOX or QMB array, pairing sensors between the QMB and MOX array, or 4 pairings from each of the MOX and QMB array. This gives four sets of processed data – MOX only, QMB only, a mix of MOX and QMB, and MOX crossed with QMB. These pairings are illustrated in Figure 5.1.

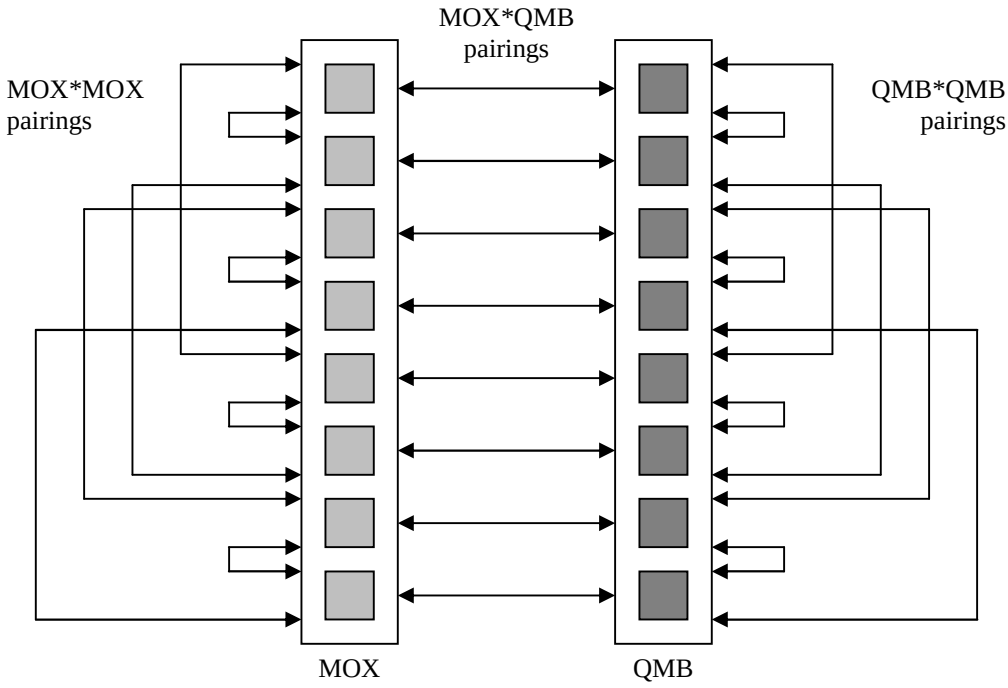


Figure 5.1 Illustration of possible sensor pairings across and between the MOX and QMB sensor arrays.

Convoluting between similar sensors is sensible for establishing any relationships that could be present in the data, however convolution between the MOX and QMB sensors is an abstract concept due to the different ways in which the sensors respond to input, and has no physical meaning. However, this idea has been pursued to see if it has value in terms of classification.

5.3.2 Feature Extraction

While the previous computational experiments only considered a single possible feature (area) from the characteristic signal, here several features will be considered. Area, peak magnitude, the product of area and magnitude, and area

divided by magnitude are chosen as potential features. This selection of features consider different forms of the overall shape of the characteristic signal, and could further help remove correlation to concentration or noise.

5.4 Classification

There are many classification approaches, which can be grouped into two broad categories: statistical (parametric) and non-parametric [5.3].

Statistical approaches represent methods that process the data in a manner compatible with statistical techniques, by assuming an underlying probability density function, such as PCA, DFA, PLS or MCR. The results for these are normally presented as graphs or charts. Statistical methods perform well with clearly separated data, but can incorrectly classify indistinct data sets due to the rigid classification criteria set after calibration [5.4].

Non-parametric methods do not assume any multinormal distributions and include artificial neural networks (ANN) and nearest neighbour classification (KNN) schemes [5.5]. Non-parametric approaches can provide a solution to more challenging datasets, such as non-linear datasets or vectors which are closer to decision boundaries.

To compare these two groups of classification approaches, a representative method from each of these two categories was chosen. A statistical principal

components analysis was chosen and a non-parametric probabilistic neural network was chosen. Both these methods are widely used within the field of artificial olfaction, and so much evidence exists as a point of comparison for this new approach.

5.4.1 Principal Components Analysis

Principal components analysis (PCA) considers data as variance in each dimension that the data possesses [5.6]. As a single data sample for this data set consists of 8 characteristic signals, the data can be considered to have 8 dimensions; to plot these data in 2- or 3-dimensional space suitable for visualisation is impossible without further processing [5.7].

A PCA processes the data, rotating it in such a way so that the maximum possible variance between data points is present in lower dimensions. While the data are no less dimensionally complex, much more of the information is contained in fewer dimensions. With a significant amount of the information now contained in 2 or 3 dimensions, a regression approach for classification is much simpler with the new axes in principal component space.

The PCA processing is carried out using the *pca* function included in the commercial software MATLAB 7.4.0 (R2007a).

5.4.2 Probabilistic Neural Network

A PNN [5.8] is constructed and trained using the *newpnn* included in the Neural Network Toolbox for the commercial software MATLAB 7.4.0 (R2007a). The neural net was defined with one hidden layer, 8 input nodes to receive the 8 sensor inputs/pairings and 4 output nodes representing the four analytes. The features extracted from the characteristic signals need to be scaled through an additional normalisation process. Auto-scaling is again used so that the features lie between 0 and 1. This is most suitable given the nature of the input data, but will make the network asymmetric (ideally the inputs would be scaled to ±1 for a symmetric network).

In order to train the PNN, the 40 samples for each analyte were split into 2 sets of 20 samples each, set A and set B. One set would be used for training the network, and the other was used as a testing vector to determine the accuracy of the network. The roles of the two sets were then reversed. This allows the construction of a confusion matrix to judge the accuracy of the neural network.

5.5 Results

The following section collects a selection of results generated from processing the MOX/QMB data set using a convolution approach, using the various

combinations of sensors and features outlined above, and classified using either a statistical or non-parametric approach. Full results can be found in Appendix C.

5.5.1 Principal Components Analysis

The following plots (Figure 5.2) show the results of the PCA regression.

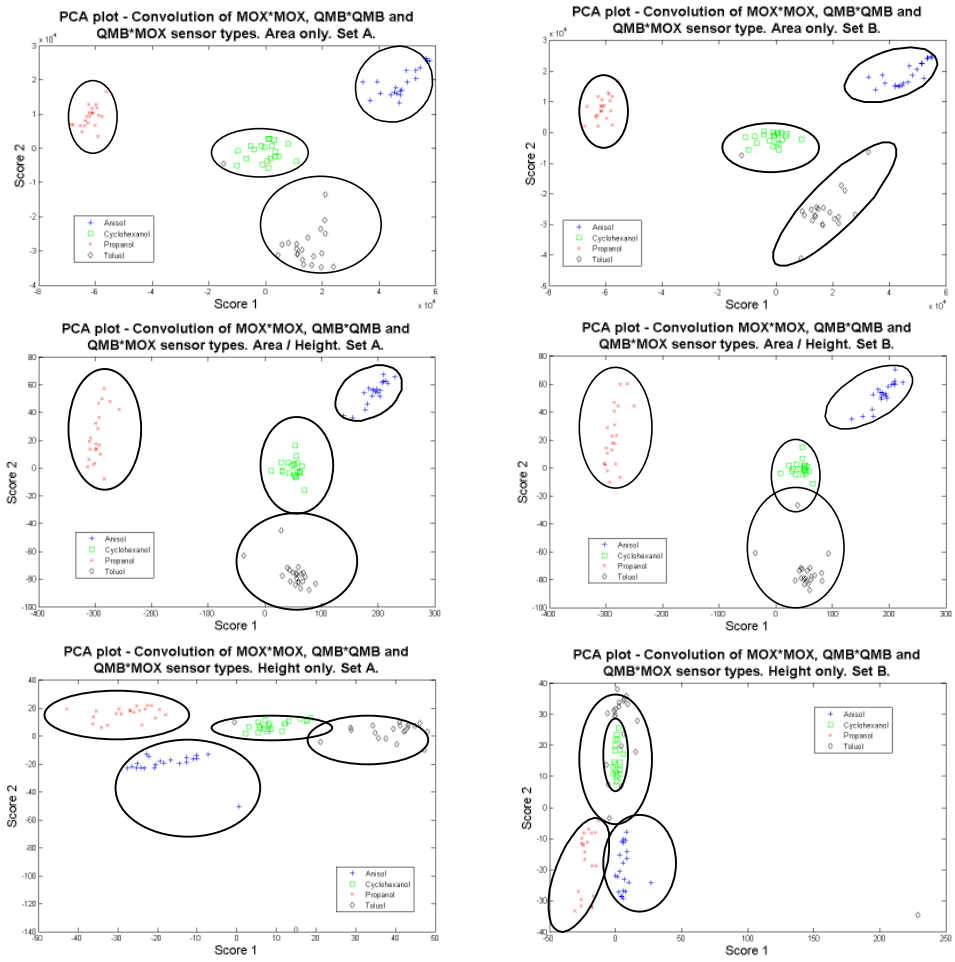


Figure 5.2 PCA plots of the first 2 principal components after convolution processing of each data set, extracting three types of feature from the characteristic signal.

These plots show the data being successfully separated when a feature including the area was used. Excluding the one clear outlier, the regression gives clear classification groupings. Only utilising the height feature was unsuccessful, with the groupings impossible to clearly separate.

5.5.2 Probabilistic Neural Network

The following tables (Tables 5.1-5.3) show the overall performance of the PNN network in classifying one data set after being trained on the other.

Table 5.1 PNN accuracy after using convolution processing using MOX*MOX, QMB*QMB and QMB*MOX sensor combinations

	Conv. Height Feature	Conv. Area Feature	Area / Height Feature
Set A Train, Set B Test	96.43%	97.62%	100%
Set B Train, Set A Test	97.62%	98.81%	100%

Table 5.2 PNN accuracy after using convolution processing using MOX*MOX and QMB*QMB sensor combinations

	Conv. Height Feature	Conv. Area Feature	Area / Height Feature
Set A Train, Set B Test	97.62%	98.81%	100%
Set B Train, Set A Test	97.62%	98.81%	100%

Table 5.3 PNN accuracy after using convolution processing using MOX*QMB sensor combinations

	Conv. Height Feature	Conv. Area Feature	Area / Height Feature
Set A Train, Set B Test	94.05%	96.43%	100%
Set B Train, Set A Test	95.24%	97.62%	100%

Table 5.4 PNN accuracy after using convolution processing using MOX*MOX sensor combinations

	Conv. Height Feature	Conv. Area Feature	Area / Height Feature
Set A Train, Set B Test	96.43%	96.43%	97.62%
Set B Train, Set A Test	97.62%	97.62%	97.62%

Table 5.5 PNN accuracy after using convolution processing using QMB*QMB sensor combinations

	Conv. Height Feature	Conv. Area Feature	Area / Height Feature
Set A Train, Set B Test	52.38%	85.71%	97.62%
Set B Train, Set A Test	61.90%	95.24%	100%

These results show that the area/height feature performed at a very high accuracy, and was clearly superior to the other features.

These results will now be discussed in detail.

5.6 Analysis of Results

From the PCA plots, it is clear that a height feature extracted from the characteristic signal is unsuitable for classification of this data set, and is supported by the lower accuracy of the PNN network when also using this feature. Area and area/height are much more successful in separating the data. However, a few outliers are less distinct in the area feature. In the area/height feature plot, these outliers are much closer to their parent group and would result

in better classification performance. This is again supported by the PNN network results, with the area feature achieving a good 90+% average accuracy, but the area/height feature achieving close to 100% overall.

Using only the results from the QMB sensor array were clearly the weakest when used alone for classification, although the area/height feature still proved to be a very good classifier in this worst case. Including the MOX sensor data in any form greatly boosted the quality of results. The most accurate classifier was the convolution of unmixed pairs of MOX and QMB data, suggesting that both arrays are bringing unique and different information to the classification problem, even though the time constants for the different sensor types are distinct and represent different physical processes.

5.7 Conclusions

In this chapter, it has been shown that the convolution processing method has been successfully applied to an available real experimental data set from standard e-nose devices (MOX and QMB), and that an improved and successful classification can be carried out. A selection of features which can be extracted from the characteristic signal were explored to further improve the performance of the convolution processing method, and while area continued to provide successful solutions to classification problems, a ratio of area and height was shown to improve the results to 100% accuracy under the right conditions.

5.8 References

[5.1] ISOCS, https://www.olfactionsociety.org/, Data originally from Tubingen University, Germany from the group of Dr. U. Weimar

[5.2] MultiSens, JLM Innovation, http://www.jlm-innovation.de

[5.3] M. Bicego *et al*., "A comparative analysis of basic pattern recognition techniques for the development of small size electronic noses", *Sensors and Actuators B*, vol. 85, pp. 137-144, 2002

[5.4] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, New York; Chichester: John Wiley & Sons, 1992

[5.5] D. J. Sheskin, *Handbook of Parametric and Non-Parametric Statistical Procedures 3rd Edition*, Boca Raton; London: Chapman & Hall/CRC Press, 2004

[5.6] A. A. Afifi and V. Clark, *Computer-Aided Multivariate Analysis* 3rd Edition, London: Chapman & Hall, 1996

[5.7] A. A. Afifi and S. P. Azen, *Statistical Analysis: A Computer Oriented Approach 2nd Edition*, New York; London: Academic Press, 1979

[5.8] L-M. Fu, *Neural Networks in Computer Intelligence*, New York; London: McGraw-Hill, 1994

Chapter 6: Parametric Performance of the Convolution-Based Method on Real Chemical Sensor Data: Quantification

6.1 Introduction

Classification is not the only identification problem that needs to be tackled. Quantification of a substance is often also important. Identifying concentrations of a chemical has important applications in control systems [6.1], and also forms an integral part in the classification of complex odours, where differing concentrations of a combination of substances can lead to different odours [6.2]. This also forms an important step to the goal of taint detection or quality control applications for an advanced odour detection device.

6.2 Experimental Data Set

A data set kindly provided by Prof. K. Persaud of Manchester University [6.3] as part of ISOCS [6.4] was to be used for quantification testing. The data collected contains information regarding data collection of a commercially supplied apple concentrate at 4 different batch concentrations: 28 μg/litre, 30 μg/litre, 33

μg/litre and 36 μg/litre, each in a water solvent. Each concentration was sampled 8 times by an array of 32 polymer sensors, extracted from a 40 ml headspace from 25 ml of solution. The number of water washes between repeats was 2. The data were supplied in .CSV format, and transferred to MATLAB 7.4.0 (R2007a) software for processing.

6.3 Applying the Convolution Method to the Data Set

Having been imported into MATLAB 7.4.0 (R2007a), the data were processed using the convolution method detailed in Appendix A. Pre-processing would not be carried out, as the concentration information would be lost through normalisation. The 32 sensors were paired for the convolution in an expanded version of the pairing scheme used in Chapter 4 (Fig 4.7).

Following processing, features would be extracted from the characteristic signals generated by the convolutions. These features would be peak magnitude, area, magnitude × area and area / magnitude.

With the feature data sets established, the data would be quantified by a partial least-squares regression approach due to the non-linear nature of the data, allowing the data to analysed in a linear nature.

6.4 Quantification

Quantification requires different statistical techniques in order to correctly differentiate between analytes. Partial least-squares regression [6.5] is a popular technique for quantification of data. This is a technique that combines linear regression with predictor variables, making it more useful for the quantification of continuous values like concentration, rather than the discrete groups required of classification [6.6].

Least-squares regression fits data to minimise the error in a least-squares sense, producing a line of regression to approximate the data [6.7],

$$Y = AX + B \tag{6.1}$$

where Y is the output independent variable, X the input data, B the regression constants and A the regression co-efficients. However, such a technique can be greatly affected by noise and is unsuitable for non-linear data. Partial least-squares regression (PLS) decomposes the input data and independent variable into a lower dimensional space before performing least-squares regression on the decomposition variables [6.8]. The decomposition acts to remove variance due to noise, ensuring the regression is performed based on the variance of the variable data. This decomposition is performed as follows,

$$X = TP^T + B \tag{6.2}$$

$$Y = UQ^T + C \tag{6.3}$$

$$U = AT + C \tag{6.4}$$

where C is the decomposition constants, U and T are decomposition co-efficients and P and Q are the decomposition variables. This decomposition results in the final regression model shown in Equation 6.5.

$$Y = (PAQ^T)X + C \tag{6.5}$$

In order to carry out a PLS, P and Q need to be estimated. The algorithm chosen to do this was a Non-linear Iterative Partial Least-Squares (NIPLS) approach, an iterative approach to the estimation of P and Q. A MATLAB implementation of this approach is available from [6.9]. This algorithm would be used to perform PLS on the data set.

A score was then generated from the PLS decomposition (Eq. 6.5), and plotted against the concentration. Each set of 8 samples are illustrated as a mean of the samples, with error defined by their variance.

6.5 Results

The following section presents the results generated from processing the concentration data set provided by Manchester University [6.3].

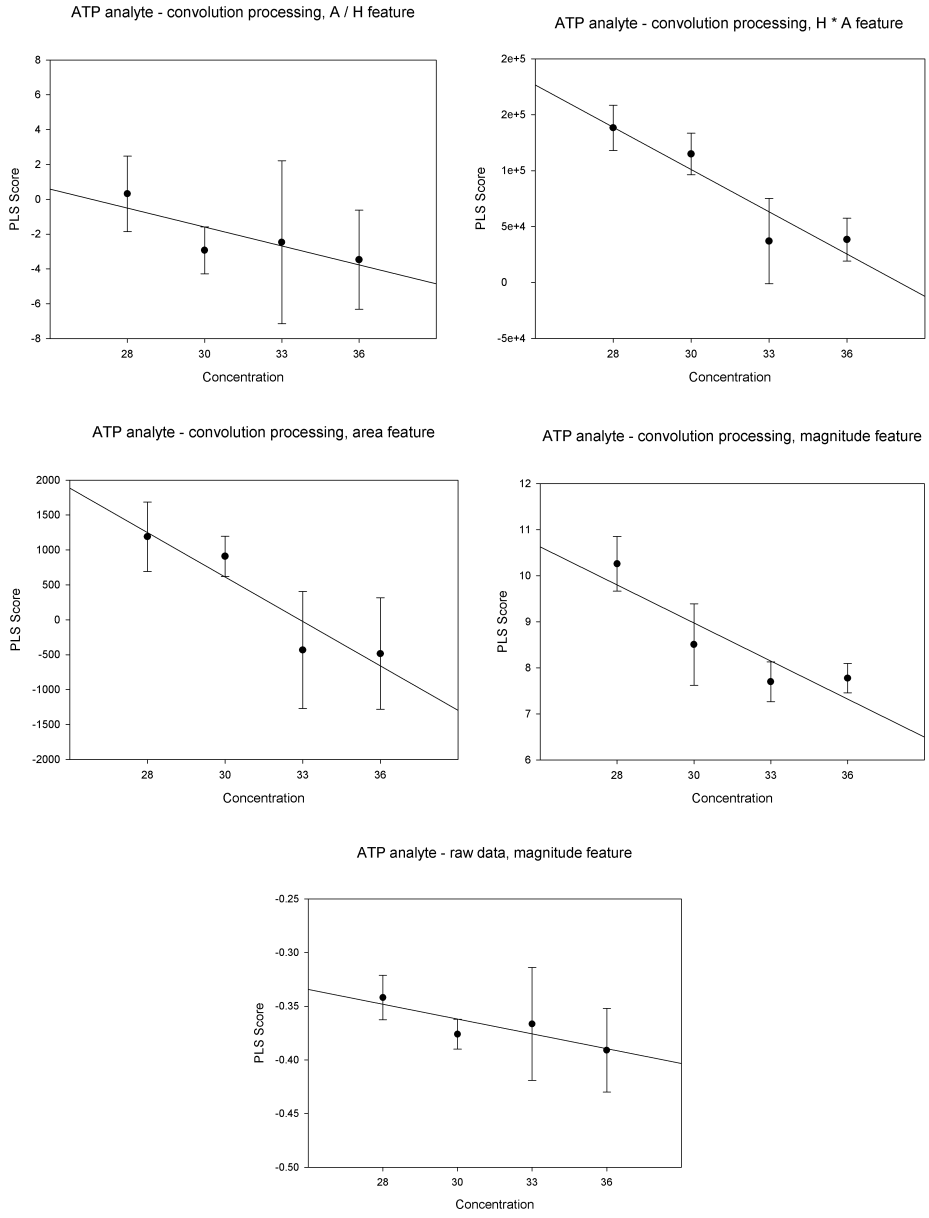


Figure 6.1 – PLS of concentration information from APT data collection. Large scale variations occur due to the nature of numerically computing convolution and area in the MATLAB software.

6.6 Analysis of Results

From Figure 6.1, it can be seen that the regression is very variable. With a dominant area feature selection after convolution (A, A/H), the error in the results is very large, making measurement of concentration using this method problematic, as most samples extend across the entire region of regression. With the magnitude feature dominant (H, H×A), the results were much improved, with the error regions to the regression being much smaller, and a steeper gradient on the line of regression. However, the error regions still remain relatively large over this period of regression. The magnitude features show a marked improvement over the regression of the raw data, where the area features show a markedly larger error area, despite the steeper gradient of the line of regression.

6.7 Conclusions

In this chapter, the convolution method has been applied to a quantification of concentration problem with some success. While the overall quality of the results is disappointing, the convolution method has shown a marked improvement over just utilising raw data for quantification purposes. It is clear from these results that a feature that focuses on magnitude is preferable for this purpose, in opposition to the area feature most suitable for classification.

6.8 References

[6.1] P. T. Moseley and B. C. Tofield, *Solid State Gas Sensors*, Bristol: Adam Hilger, 1987

[6.2] M. J. Madou and S. R. Morrison, *Chemical Sensing with Solid State Devices*, New York: Academic Press, 1989

[6.3] Prof. K. Persaud, Manchester University, Personal Communication

[6.4] ISOCS, https://www.olfactionsociety.org/

[6.5] V. Esposito Vinzi, W. W. Chin, J Henseler and H. Wang, *Handbook of Partial Least Squares*, Berlin; New York: Springer, 2010

[6.6] A. A. Giordano and F. M. Hsu, *Least Square Estimation with Applications to Digital Signal Processing*, New York; Chichester: John Wiley & Sons, 1985

[6.7] P. Gans, *Data Fitting in the Chemical Sciences: By the Method of Least Squares*, Chichester: John Wiley & Sons, 1992

[6.8] Å. Björck, *Numericla Methods for Least Squares Problems*, Philadelphia: SIAM, 1996

[6.9] Y. Cao, MATLAB Central – File Exchange – Partial Least-Squares and Discriminant Analysis - http://www.mathworks.com/matlabcentral/fileexchange/18760-partial-least-squares-and-discriminant-analysis (last accessed 01/08/2010)

Chapter 7: Application of the Convolution Method to Simulated Data from an Artificial Olfactory Mucosa

7.1 Introduction

Advancements in the field of artificial olfaction continue apace. As our understanding of the biological olfactory system increases, so devices are developed to take advantage of this new knowledge. These avenues led to the categories of devices discussed in the previous chapters, and the processing methodologies developed alongside them [7.1, 7.2]. However, new devices are constantly under development, taking advantage of the newest understandings of the process of olfaction. One such device is the artificial olfactory mucosa [7.3]. Here a novel device being developed is outlined, and based on simulations developed for it, application of the convolution processing method to the artificial mucosa is performed.

7.2 Artificial Olfactory Mucosa Simulation

The artificial olfactory mucosa (AOM) is a concept developed and constructed at the University of Warwick [7.3]. The device takes advantage of the nasal chromatography effect [7.4, 7.5]. The AOM is embodied in a simple form here as multiple points along the retentive column in a single chromatograph (and later developed with multiple columns and arrays). This produces a more complex and information rich spatio-temporal data set, as the data now have multiple frames of reference for successive sets of temporal data. This information-rich spatio-temporal signal is not utilised by current data processing techniques in use in the field of artificial olfaction, and so the development of new data processing methods is needed in order to fully exploit this new type of data [7.6].

7.2.1 Outline of Artificial Olfactory Mucosa Simulation

The simulation developed during the development of the Warwick Protonose III device [7.3] forms the basis of the e-mucosa simulation, and a similar approach to generating data suitable for processing found in Chapter 4 is used here. The same two arrays of sensors, the first 10 and final 10, form the AOM device, but are now considered to be a single device rather than two distinct devices (e-nose and c-nose).

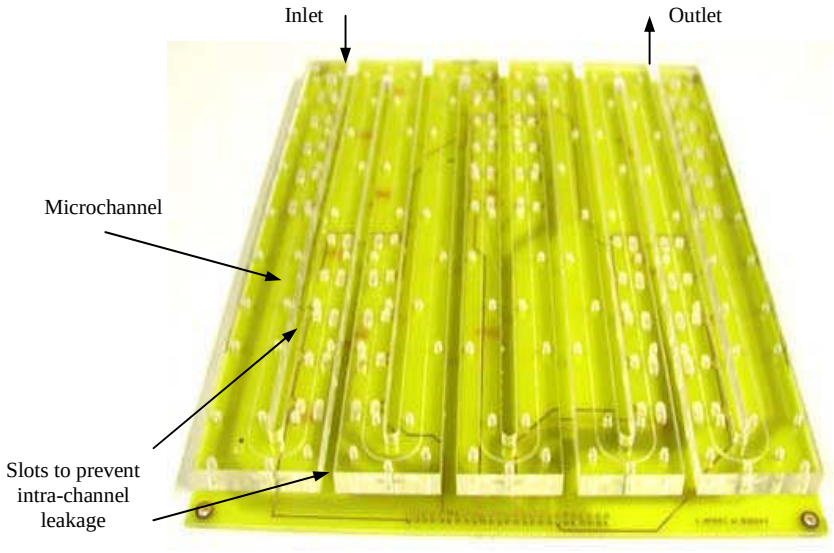


Figure 7.1 Physical realisation of the Protonose III [7.3]

As a single ideal simulation was carried out to demonstrate the viability of the device, this signal will form the basis of the processing data set, and a noise simulation performed to produce a group of varied signal sets suitable for a processing problem.

7.2.2 Noise Simulation

The noise simulations are identical to those carried out in Chapter 4, and the equations and processing detailed there is also applied to the simulated AOM device. It should be noted that as a single device, it was determined that the baseline drift of signals from sensor pairs would be similar, because the sensors would be subject to the same conditions and experience. Thus, the direction of the drift for a given sensor pair would always be the same. The magnitude of the noise was chosen to be 5%. For additive and sample noise, this is typical. However, this remains a very large value for drift-type noise, as it is assumed to

occur between samples and over the testing period (100 seconds). A more realistic value would be 0.5% or less.

7.2.3 Final Simulation Data Set

The final data set was simulated based on the simulations generated from the e-mucosa [7.3], using the commercial software MATLAB 6.0.0.88 (r12) and the FEMLAB 2.3b finite element model [7.7]. This simulation generated 3 data sets, simulating a 5 second pulse of each of ethanol, toluene and a 50/50 mixture in air of the two passing through the system. The retentive micro-channel simulated has a cross-sectional geometry of 0.5 mm × 0.5 mm and a length of 2.3 m, and coated with Parylene C as the stationary phase. 40 sensors, in 4 groups of 10 differing tunings, are arranged along the micro-channel as shown in Figure 7.1. For this simulation, the AOM e-mucosa device is formed from the first 10 sensors, each paired with their counterpart in the final 10 sensors.

These pre-generated signals were used as a base for generating noisy signals in MATLAB 7.4.0 (R2007a). From each of these 3 data sets, 20 random data sets were generated by the addition of a noise signal category, producing a complete data set of 60 samples containing 3 analytes. This produces 3 60-sample data sets, one for each of the noise categories; additive noise, sample variance and baseline drift. A further 3 data sets were generated using binary combinations of noise categories; additive and sample variance, additive and baseline drift and sample variance and baseline drift. For all of these data sets, the noise value was set to a 5% level. The direction of the baseline drift noise between a pair of similar

sensors was fixed to be the same in all cases. The result of this data generation results in 6 noisy e-mucosa data sets, totalling 120 data samples.

7.3 Processing the Simulation Data

The simulated data sets can now be processed, and a similar methodology as that presented in previous chapters will be used. A pre-processing stage rectifies any potential bias in the data set, ensuring each sensor provides an appropiate weight to the processing process. The main processing stage then occurs, where the data are processed into characteristic signals. From these characteristic signals, a feature suitable for the chosen classification approach can be extracted. The processing was performed in the commercial software MATLAB 7.4.0 (R2007a), and details of function implementation can be found in Appendix A.

7.3.1 Pre-processing

The pre-processing strategies outlined in Chapter 4 are also applied here. These are the auto-scaling (equation 4.14) and auto-ranging (equation 4.15) normalisation approaches, as well as simply using the raw data (equation 4.16) for processing. This highlights the need to normalise the data to ensure that no sensor is being uncharacteristically weighted by the system.

7.3.2 Processing

The AOM device generates spatio-temporal data, with data produced by the two sensor arrays related in terms of spatial displacement and temporal displacement. As such, regular processing methods do not make use of this related nature of the data. In order to take advantage of this property, a new approach is required. The convolution approach, which combines the properties of two different signals, is suggested as being suitable for this purpose. The relationships between each pair of signals will be borne out in the resulting characteristic signal, highlighting the differing effect that the retentive channel has on different analytes.

Of course, convolution is not the only manner in which two different time series signals can be combined. Other methods of producing characteristic signals will also be considered: a signal product and a difference signal.

7.3.2.1 Convolution Characteristic Signal

The convolution method involves performing a convolution between two signals. These signals are chosen from a pair of similar sensors, one from each of the two arrays, S_A and S_B,

$$y^{c}_{AOM}(t) = \int_{\tau=-\infty}^{\infty} S_B(\tau) S_A(t-\tau) d\tau \qquad (7.1)$$

generating the convolution characteristic signal, y^{c}_{AOM}.

7.3.2.2 Product Characteristic Signal

The product signal is generated from the product of each of the sensor signal pairings, S_A and S_B. In order to maintain the signal's nature and scale of magnitude, this is preformed as a signed root-product,

$$y^{p}_{AOM}(t) = \mathrm{sgn}\left(S_A(t)S_B(t)\right)\sqrt{\left|S_A(t)S_B(t)\right|} \tag{7.2}$$

generating the product characteristic signal y^{p}_{AOM}.

7.3.2.3 Difference Characteristic Signal

The simplest form of characteristic signal would be a simple difference signal between the two related sensors, S_A and S_B. The decayed second signal is subtracted from the first signal to produce a difference characteristic signal, y^{d}_{AOM},

$$y^{d}_{AOM}(t) = S_A(t) - S_B(t)\,. \tag{7.3}$$

7.3.3 Feature Extraction

These three approaches each produce an array of 10 characteristic signals. From this array of signals, a feature is chosen to produce a feature array suitable for use

in the classification approaches. For this simulation, the area under the curve is chosen as the feature to be extracted.

7.4 Post-Processing and Classification

Once a suitable feature has been selected, the feature sets can now be used in classification. The processing has resulted in 54 feature sets (6 noise types × 3 pre-processing methods × 3 characteristic signal types). These feature sets first have a principal components analysis applied to them. Classification regions can then be defined through linear regression. To produce a quantitative measure of the quality of the processing approach, the principal components sets have a t-test preformed on them.

7.4.1 Principal Components Analysis

A PCA is applied to the processed data sets, using the procedure outlined in Chapter 4. The PCA transforms the data so that a maximum amount of variance is present in as few principal components as possible. If the majority of the variance is contained in 2 or 3 principal components, this makes the PCA results suitable for graphical presentation, shows that the sensors are linearly correlated, and so suitable for classification groups to be defined by linear regression. For purposes of this simulation, linear regression will not be performed, but the PCA provides means for a graphical representation of the processed data. The amount

of variance contained in the first 2 principal components was shown to be 95%+ in all cases here, so the results of the PCA were plotted using the first 2 principal components. A selection of plots are presented in figure 7.2 (the full set of PCA plots can be found in Appendix D).

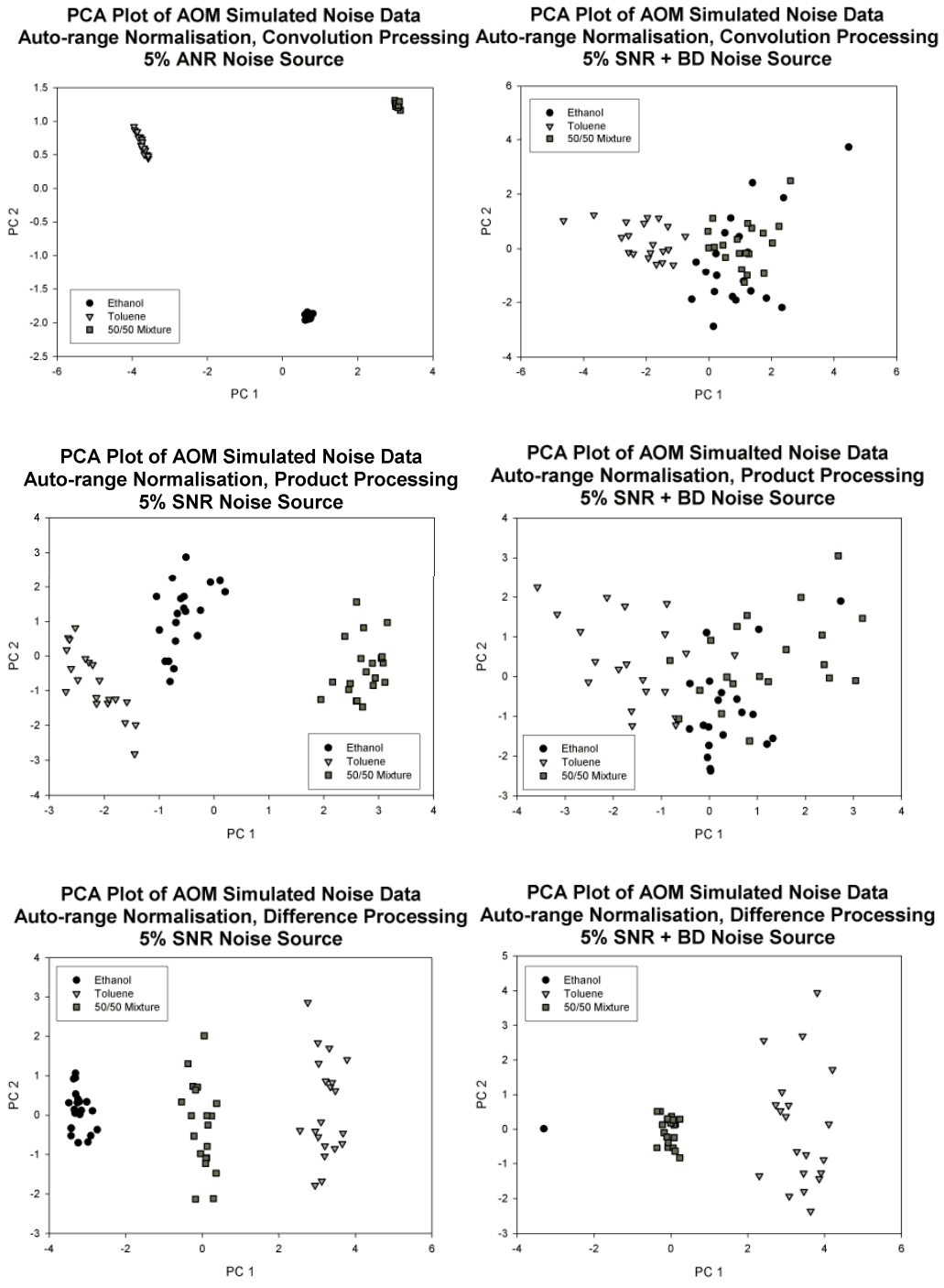


Figure 7.2 PCA plots of AOM simulation data sets with added noise.

7.4.2 t-test

A t-test is performed on the principal component data sets to provide a quantitative measure for the quality of data separation in the PCA plot, using the method outlined in Chapter 4, generating t-scores between pairs of analytes for the first two principal components, using a separate variance t-test performed in Unistat 5.5 software. This gives a measure by which the various processing techniques can be compared to each other. Tables 7.1-7.3 summarise the t-test results, and include results from Chapter 4 for the e-nose and c-nose for comparison.

		t-scores									
		Convolution		Root-Product		Difference		E-Nose		C-Nose	
		PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2
SV	Eth-Tol	118.30	93.02	167.43	231.76	71.44	0.15	198.13	3.86	146.08	6.08
	Eth-Mix	55.55	86.31	234.60	138.44	36.64	0.39	121.97	35.96	98.14	24.77
	Tol-Mix	125.52	3.02	344.81	93.24	37.80	0.67	87.53	46.31	100.70	54.52
SNR	Eth-Tol	491.61	75.34	361.60	82.38	129.86	0.76	370.43	3.91	352.92	13.83
	Eth-Mix	176.20	196.18	49.27	198.39	54.56	7.74	214.59	40.94	239.27	59.75
	Tol-Mix	407.09	140.97	299.26	107.93	43.57	7.93	226.98	91.91	227.79	103.27
BD	Eth-Tol	14.74	0.58	15.99	1.61	68.43	0.26	226.20	3.52	172.90	6.20
	Eth-Mix	6.95	1.72	2.98	4.16	25.45	0.65	141.15	39.58	131.88	27.45
	Tol-Mix	12.34	1.28	10.28	2.59	25.19	0.26	123.10	59.09	81.63	50.81
SV+SNR	Eth-Tol	573.20	83.51	393.45	61.67	119.42	1.11	405.74	4.20	400.96	16.18
	Eth-Mix	171.11	211.01	33.19	190.01	50.65	7.65	243.05	43.91	289.17	70.47
	Tol-Mix	489.25	140.16	365.20	79.26	39.52	7.63	243.54	96.90	222.83	123.24
SV+BD	Eth-Tol	18.49	2.34	19.43	1.86	59.32	0.08	231.34	2.39	234.48	7.33
	Eth-Mix	6.94	5.42	5.37	3.78	27.29	0.14	130.73	25.58	177.18	35.46
	Tol-Mix	15.84	3.66	10.36	2.25	26.63	0.08	107.21	52.78	90.39	52.51
SNR+BD	Eth-Tol	12.17	0.71	16.22	1.72	73.84	0.07	237.46	3.05	193.47	6.66
	Eth-Mix	4.72	4.76	6.11	8.16	31.07	0.96	131.48	28.77	122.83	30.77
	Tol-Mix	11.06	0.11	11.93	6.26	27.09	0.94	145.18	56.94	107.90	46.75

Table 7.1 t-scores generated between each pair of analyte groups for all the processed feature sets, using a separated variance t-test using an auto-scaling pre-processing technique

		t-scores									
		Convolution		Root-Product		Difference		E-Nose		C-Nose	
		PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2
SV	Eth-Tol	126.37	80.27	221.82	298.78	62.45	0.24	190.65	5.36	179.78	7.71
	Eth-Mix	68.05	72.04	170.79	161.23	29.06	0.28	124.48	29.22	114.79	35.01
	Tol-Mix	203.07	11.54	261.47	63.84	28.06	0.44	122.13	59.62	121.00	49.83
SNR	Eth-Tol	164.3	82.32	13.13	6.40	77.49	0.14	350.36	11.21	439.10	13.77
	Eth-Mix	140.83	283.91	30.96	5.26	44.42	1.50	221.48	55.08	280.99	55.93
	Tol-Mix	265.98	18.96	40.68	1.54	39.08	1.21	196.47	93.29	196.76	116.80
BD	Eth-Tol	6.82	0.61	3.41	0.72	49.44	0.56	191.49	4.34	225.96	7.96
	Eth-Mix	2.44	0.62	4.85	2.99	78.29	2.15	121.95	23.21	161.95	30.95
	Tol-Mix	11.41	0.00	8.61	0.80	23.12	0.17	97.39	46.07	97.02	75.88
SV+SNR	Eth-Tol	186.64	96.99	15.55	0.05	86.08	0.31	336.98	9.15	492.66	16.75
	Eth-Mix	106.75	217.90	25.69	0.10	39.03	0.09	219.05	47.18	333.96	76.44
	Tol-Mix	283.54	21.71	41.03	0.10	38.77	0.17	207.78	101.34	238.75	128.46
SV+BD	Eth-Tol	10.45	2.95	4.78	4.90	66.50	0.03	221.32	6.20	185.00	7.16
	Eth-Mix	5.06	5.63	8.87	5.76	67.34	0.93	138.47	33.67	123.64	31.16
	Tol-Mix	15.02	0.93	12.94	0.48	30.09	0.09	117.00	51.24	91.64	46.28
SNR+BD	Eth-Tol	9.34	1.91	7.26	3.48	554.17	0.07	199.73	6.21	214.46	7.60
	Eth-Mix	0.37	1.49	2.42	3.45	84.67	0.60	130.69	32.35	140.49	31.03
	Tol-Mix	12.12	0.60	7.92	0.01	26.33	0.21	96.50	67.68	115.20	48.60

Table 7.2 t-scores generated between each pair of analyte groups for all the processed feature sets, using a separated variance t-test using an auto-ranging pre-processing technique

		t-scores									
		Convolution		Root-Product		Difference		E-Nose		C-Nose	
		PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2	PC 1	PC 2
SV	Eth-Tol	196.02	90.83	253.94	303.86	10.08	0.53	219.63	5.45	187.79	6.25
	Eth-Mix	64.56	84.32	224.00	152.33	3.86	0.53	128.82	29.20	130.72	25.79
	Tol-Mix	180.65	13.49	344.76	55.60	6.97	0.21	109.13	50.45	84.41	50.02
SNR	Eth-Tol	197.80	100.33	12.43	3.72	10.71	1.17	274.61	8.30	459.78	14.80
	Eth-Mix	128.49	231.94	28.19	3.79	5.51	.035	171.68	40.57	324.15	72.70
	Tol-Mix	306.29	22.05	39.98	0.99	4.46	1.12	168.90	83.82	194.35	107.66
BD	Eth-Tol	8.50	1.36	6.95	2.34	1.11	0.54	242.04	6.46	213.07	7.62
	Eth-Mix	2.66	4.21	3.03	6.28	0.33	0.11	151.31	36.27	141.55	30.18
	Tol-Mix	10.26	2.27	8.72	2.43	1.13	0.08	104.15	58.64	113.18	67.07
SV+SNR	Eth-Tol	203.95	89.48	12.99	6.50	14.15	0.62	318.04	11.79	418.27	13.44
	Eth-Mix	128.85	239.12	24.67	4.20	4.43	0.29	199.32	65.88	296.34	5921
	Tol-Mix	282.91	20.20	51.46	0.39	4.02	0.19	196.40	86.71	186.68	109.46
SV+BD	Eth-Tol	8.43	1.91	3.99	5.86	1.28	0.07	238.86	6.12	206.40	8.18
	Eth-Mix	2.23	3.99	5.80	2.24	0.09	1.61	158.14	32.62	125.29	31.28
	Tol-Mix	14.97	1.29	0.13	9.87	0.90	1.42	110.48	57.78	79.75	48.22
SNR+BD	Eth-Tol	9.19	0.19	5.06	1.15	0.44	0.34	267.40	7.25	156.73	6.09
	Eth-Mix	3.65	0.90	5.25	0.50	0.38	0.01	163.23	34.66	108.70	24.65
	Tol-Mix	13.14	0.16	9.99	1.30	0.10	0.31	97.38	42.97	100.47	60.53

Table 7.3 t-scores generated between each pair of analyte groups for all the processed feature sets, using a separated variance t-test with no pre-processing

7.5 Analysis of Results

As is evident from the PCA plots, the three analytes are linearly separable in all processing cases and combinations for the SV, SNR and SV+SNR noise sources. In these cases, t-scores show a comparable result between the AOM processing methods and the e-nose and c-nose devices. The t-scores for PC1 are similar throughout, and the t-scores are increased in PC2 for the AOM processing approaches. This shows a similar level of separability in the first dimension, but increased separability in the second dimension for the AOM device.

However, where BD noise sources are present, the AOM processing approaches are markedly weaker than the e-nose and c-nose results, which maintain their performance across the board.

The difference characteristic signal for the AOM is shown to be the weakest overall, although it does not show the weakness to baseline drift that the convolution or root-product characteristic signal. The difference approach also produced the most anomalies during testing (such as Table 7.2, SNR+BD, PC1), highlighting it as the least reliable set of results.

The pre-processing methods had relatively little impact on the shape of the results, but the auto-ranging and auto-scaling did boost the performance of the AOM processing approaches.

7.6 Conclusions

The results highlight a weakness to excessive baseline drift (5% level) when processing data from an AOM device. This indicates that such a device will be sensitive to drift, but relatively insensitive to sample variance or SNR noise sources. Such a device would need to be well maintained to ensure the device's baseline does not alter, but would work well in a general day-to-day environment. However, the poor responses to drift are not a real problem, as the system was subjected to a very high level of drift (a ±5% noise impact between samples) – drift noise is usually much lower than this for individual samples, less than 0.5% over a typical measurement period of 1-2 minutes, and is more representative of a long-term systematic drift or major alteration to environmental conditions.

In comparison to the e-nose and c-nose results, the convolution characteristic signal performs very well in response to both SV and SNR noise. The increased separability in the second dimension, without compromising the first dimension, shows that this method has an increased sensitivity, even in this highly idealised simulation. The root-product method performs well also, although is more sensitive to SNR noise, making convolution processing the overall better choice. The difference approach, although not suffering from any particular weaknesses, is not suitable here due to its overall lower performance.

The different pre-processing methods do not really impact the results, although a small improvement could be observed over the raw data. The lack of difference

between auto-scaling and auto-ranging was expected, as the source signals were positive, thus both normalisation approaches would effectively be the same in this case. However, the spread in signal magnitudes is modest and serves to limit the impact of pre-processing.

7.7 References

[7.1] J. W. Gardner and P. N. Bartlett, A Brief History of Electronic Noses, *Sensors and Actuators A*, vol. 18 (1994), pp. 211-210

[7.2] T. C. Pierce, S. S. Schiffmann, H. T. Nagle and J. W. Gardner, *Handbook of Machine Olfaction: Electronic Nose Technology*, Weinheim: Wiley-VCH, 2003

[7.3] S. L. Tan, "Smart Chemical Sensing: Towards a Nose-on-a-Chip," Ph.D. dissertation, School of Engineering, University of Warwick, Coventry, UK, 2005

[7.4] M. J. Serby and K. L. Chobor, *Science of Olfaction*, New York; London: Springer-Verlag, 1992

[7.5] A. K. Wanekaya *et al.*, "Multicomponent analysis of alcohol vapors using integrated gas chromatography with sensor arrays", *Sensors and Actuators B*, vol. 110, pp. 41-48, 2005

[7.6] M. Bicego *et al.*, "A comparative analysis of basic pattern recognition techniques for the development of small size electronic noses", *Sensors and Actuators B*, vol. 85, pp. 137-144, 2002

[7.7] S. L. Tan, J. A. Covington, J. W. Gardner, T. C. Pearce, "Finite element simulation of a biomimetic olfactory microsystem for spatio-temporal signal generation", *AsiaSim 2007*, Springer, 2007, pp. 216-226

Chapter 8: Dual-Channel Artificial Olfactory Mucosa: Spatio-Temporal Signal Processing

8.1 Introduction

In this chapter, everything learned from the previous chapters is applied to an advanced artificial olfactory mucosa under development at Warwick University. This work was undertaken in parallel with another PhD research student, Fauzan Che Harun, who developed the device as part of his work as a member of the Sensors Research Laboratory, Warwick University, UK [8.1-8.3].

8.2 The Artificial Olfactory Mucosa: Dual-channel Large Array Electronic Nose

The device under development is very different to existing artificial olfactory systems. It is composed of three large electronic nose arrays, each containing 300 sensors of 24 different tunings (12 per tuning). One array samples the input immediately, after which the input is split along two retentive channels, one with a polar retentive material and the other with a non-polar retentive material. After

these two channels are the second and third electronic nose arrays in a parallel configuration (Figure 8.1) [8.1]. This embodiment goes beyond the human olfactory system which has an aqueous (polar) coating. However, olfactory binding proteins are used to capture hydrophobic compounds and transport them to the olfactory receptors; a process known as LUSH. In this way, non-polar molecules can be detected by the olfactory mucosa.

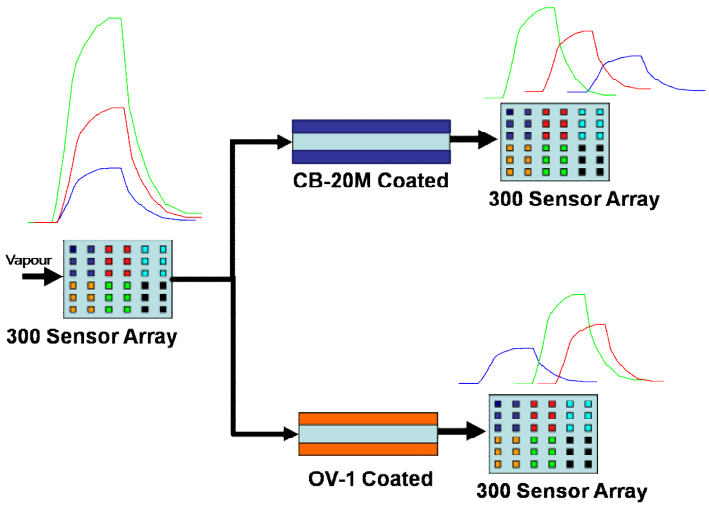


Figure 8.1 Dual-channel Large Array Electronic Nose [8.1]

The retentive channels have a cross-section of 0.38 mm × 0.25 mm and are 2.0 m in length. The first column is coated in a 5 μm thick layer of OV-1 (a non-polar stationary phase) and the second column is coated in a 5 μm of Carbowax 20M (a polar stationary phase). These different stationary phases have a differing retentive effect on the components of the analyte as it passes along the channel, producing a data set with spatio-temporal properties as the two sensor arrays produce different signals in the same time frame.

The sensors are arrayed in a 25 × 12 matrix, arranged as shown in Figure 8.2. The tunings of the sensors are listed in Table 8.1. The polymer composite solution for each sensor tuning was prepared by dissolving the polymer blend in 20 ml of solvent, followed by the addition of the carbon black (Cabot corp. USA). Each tuning was deposited using an airbrush and mask with twelve 200 μm holes arranged in a 4 × 3 matrix. The sensor resistance was set during deposition to lie between 1 kΩ and 5 kΩ.

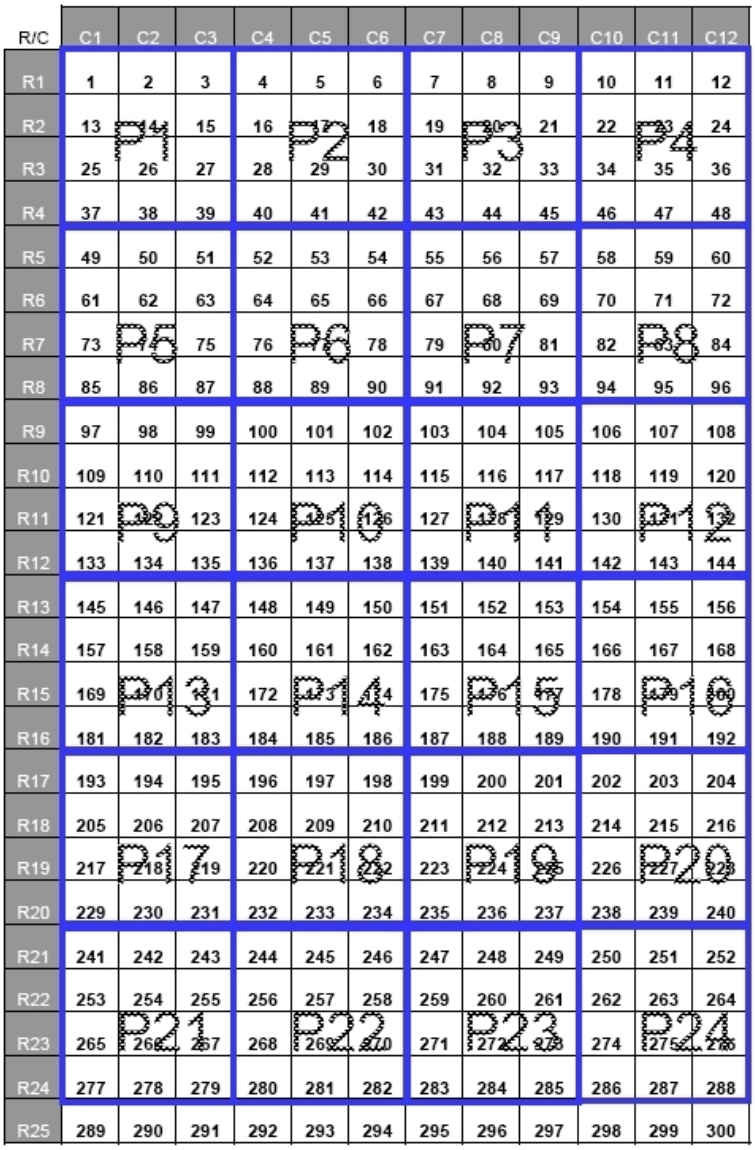


Figure 8.2 Diagram of the e-mucosa sensor and pattern of sensor tunings [8.1]

No.	Polymer Composite	Polymer A (g)	Polymer B (g)	Carbon black (g)	Solvent (20 ml)
P1	Poly(stylene-co-butadiene) PSB	0.7	0	0.175	Toluene
P2	Poly(ethylene-co-vinyl acetate) PEVA	1.2	0	0.3	Toluene
P3	Poly(caprolactone) PCL	1.2	0	0.3	Toluene
P4	Poly(9-vinylcarbazole) PVC	0.7	0	0.175	Toluene
P5	Poly(ehtylene glycol) PEG	1.2	0	0.3	Toluene
P6	Poly(4-vinyl phenol) PVPH	1.2	0	0.3	Ethanol
P7	Poly(methyl methacrylate) PMM	1.2	0	0.3	Ethanol
P8	Poly(vinyl pyrrolidone) PVPD	0.7	0	0.175	Ethanol
P9	Poly(bisphenol A carbonate) PBA	0.7	0	0.175	Dichloromethane
P10	Poly(sulfane) PSF	0.7	0	0.175	Dichloromethane
P11	PSB 50% + PEVA 50%	0.35	0.6	0.2375	Toluene
P12	PSB 50% + PCL 50%	0.35	0.6	0.2375	Toluene
P13	PEVA 50% + PCL 50%	0.6	0.6	0.3	Toluene
P14	PEG 50% + PVPH 50%	0.6	0.6	0.3	Ethanol
P15	PSB 50% + PVC 50%	0.35	0.6	0.2375	Toluene
P16	PEVA 50% + PVC 50%	0.6	0.35	0.2375	Toluene
P17	PCL 50% + PVC 50%	0.6	0.35	0.2375	Toluene
P18	PMMA 50% + PSB 50%	0.6	0.35	0.2375	Toluene
P19	PMMA 50% + PEVA 50%	0.6	0.6	0.3	Toluene
P20	PMMA 50% + PCL 50%	0.6	0.6	0.3	Toluene
P21	PMMA 50% + PVC 50%	0.6	0.35	0.2375	Toluene
P22	PEG 50% + PVPD 50%	0.6	0.35	0.2375	Ethanol
P23	PVPH 50% + PVPD 50%	0.6	0.35	0.2375	Ethanol
P24	PBA 50% + PSF 50%	0.35	0.35	0.175	Dichloromethane

Table 8.1 Details of polymer compositions used for AOM sensors [8.1]

With this device, information relating to the temporal, spatial and polarity of an analyte is collected by the three sensor arrays; the spatial differences between the different sensor tunings, the temporal differences between the pre-channel and post-channel arrays, and the polar differences between the polar-retentive array and the non-polar retentive array, as defined by the linear solvation energy relationship (LSER) equation (8.1) [8.4]. The solvation energy (k) is defined by

$$\log k = c + rR_2 + s\pi_2^H + a\alpha_2^H + b\beta_2^H + l\log L^{16} \tag{8.1}$$

where the parameters R_2, π_2^H, α_2^H, β_2^H and L^{16} characterised by the analyte, and the coefficients *c*, *r*, *s*, *a*, *b* and *l* dependant on the stationary phase coating. In characterising the analyte, the parameter R_2 is a factor based on the polarisability of the *n*- and π-electrons; π_2^H the dipolarity; α_2^H the hydrogen-bond acidity; β_2^H the hydrogen-bond basicity; and L^{16} the gas-liquid partition coefficient in hexadecane. In characterising the stationary phase coating, *c* is the constant representing the basic partition of an analyte with no other interactions as described by the other parameters; *r* describes the polarisability; *s* the dipolarity; *a* the hydrogen-bond basicity; *b* the hydrogen-bond acidity; and *l* is a factor determined by the ability of the coating to distinguish between members of a homogeneous series of compounds. [8.4]

This creates a very complex, information-rich set of sensor signals. Current processing methods would find it difficult to exploit fully the temporal or polar nature of this data, let alone the combination of all three information types. This spatio-temporal data set requires a new approach to processing, such as that proposed by the convolution processing method in Chapter 7.

8.3 Application of the Convolution Processing Method

Applying the convolution method to this system follows the same basic processes as those detailed in the previous examples. However, the increased complexity of this system introduces many more options for the combinations of signals and the way in which they can be related to one another.

The same basic data exploration is followed; using pre-processing, processing, feature selection, feature extraction, then post-processing and classification.

For pre-processing, two procedures were performed. Firstly, the data were analysed and erroneous signals censored. Due to the experimental nature of the device, many sensors were producing signals that could easily be identified as in correct (Figure 8.3 illustrates this) – such as sensors whose output signal is too high (i.e. saturated) or too low (i.e. produce negligible signals). Sensors reaching the output voltage value of 10 V or a signal with magnitude of less that 10% of other similar sensors in the group were removed. This reduces the noise element in subsequent multivariate data analysis; especially reducing the noise generated during the feature extraction process. The redundant groups of multiple sensors allow these non-functioning sensors to be removed without compromising the variety of the data set.

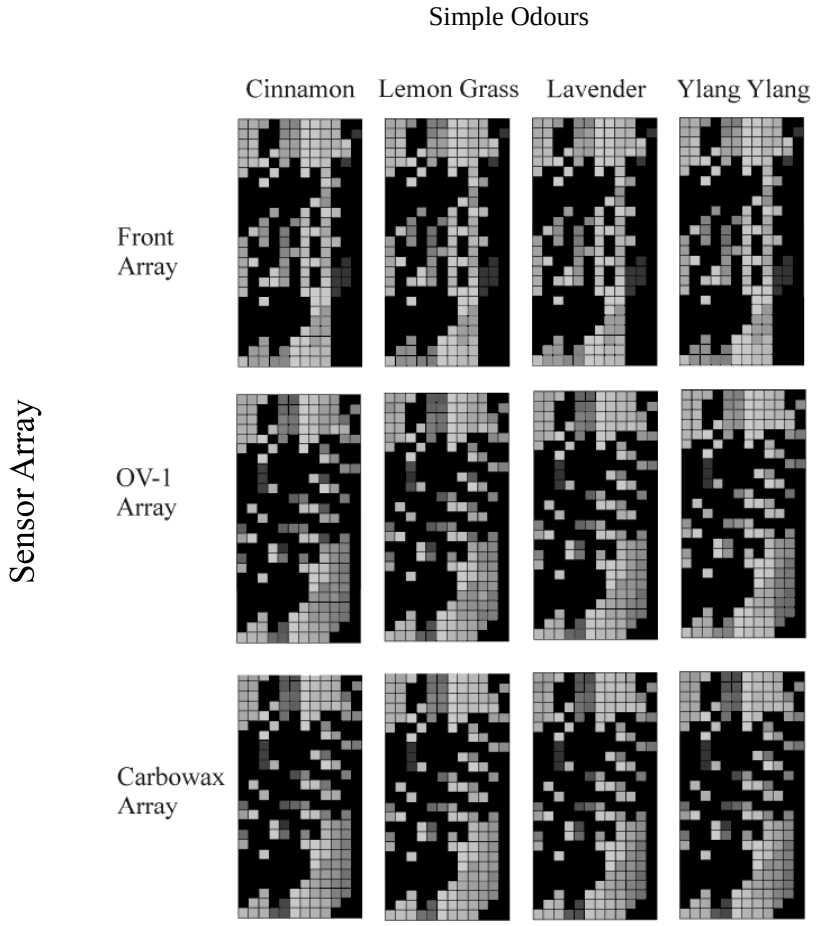


Figure 8.3 Graphical illustrations of the sensor array outputs from four essential oil analytes – black represents a 0-10% output, white saturated output and shades of grey represent a normalised peak magnitude. In this illustration of the spatial domain of the data, the outputs appear to be highly correlated for these odorants.

Sensors signals that are not censored are then subjected to an auto-scaling normalisation method.

$$S^j(t) = \frac{\left|X^j(t)\right|}{\max\left(\left|X^{\forall j}\right|\right)} \tag{8.2}$$

The convolution method can then be used for processing. The feature selected for extraction was the area under the characteristic convolution signal. However, some new approaches to classification and presentation will also be considered, in addition to principal components analysis.

In the processing stage, a number of processing pairs are available; the first array and the array after the polar channel, the first array and the array after the non-polar channel, and the two arrays after the retentive channels.

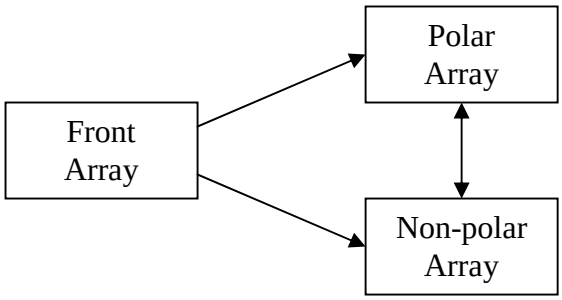


Figure 8.4 Possible convolution pairings of the 3 sensor arrays.

8.4 Classification

Multiple approaches to classification were considered for this problem. Starting with PCA, a time independent classification approach, moving onto a non-parametric approach using a probabilistic neural network (PNN).

8.4.1 Principal Components Analysis

A PCA is performed on the processed data. This time independent linear method forms a baseline for comparison to more advanced methods. The PCA is carried out as it had been in previous chapters, and the first two principle components taken and extracted for graphical presentation.

8.4.2 Probabilistic Neural Network

Due to the small sample size available from the artificial mucosa, the data are unable to be split into training and testing sets while maintain a suitable number of samples so that the classification can be statistically significant. This means using alternative ways of organising and using data. This is achieved by using a bootstrap training approach [8.5].

The bootstrap train and test approach omits a subset of the samples, trains the neural network with the remaining samples, and finally uses the initial subset as its test vectors. This is repeated for a significant number of possible unique subsets to arrive at a confusion matrix to evaluate the performance of the PNN as a classifier. The PNN [8.6] is constructed and trained using the *newpnn* function included in the Neural Network Toolbox for the commercial software MATLAB 7.4.0 (R2007a). The data are normalised to lie between 0 and 1 in order so that it can be used in the PNN.

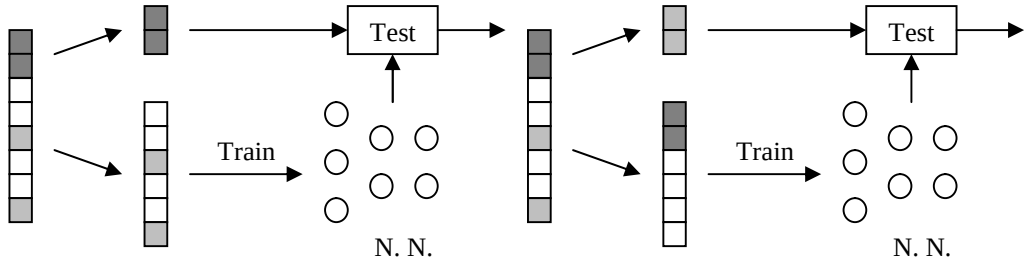


Figure 8.5 Diagram illustrating the bootstrap train-and-test approach to training a PNN

The censorship process results in an uneven distribution of sensors among the 24 types of sensors, with some groups having many sensors omitted, and other groups only losing a few sensors. Even if this was not the case, using all of these sensors may not produce the best classification performance – some may not be reacting to the analyte at all, and just contributing noise to the classification processing. In order to select the best subset of sensors to use, a Monte Carlo randomiser [8.7] was used during the training process. This arranges the sensors in a random order and performs a training cycle using just the first sensor. The second sensor is added to the training vector, and the training cycle repeated. If the second sensor has improved the classification performance, the sensor is retained – if not, the sensor is discarded. This is repeated through the whole list of sensors until the best performing subset has been selected. This is a sub-optimal non-exhaustive feature selection method, but has been shown to produce reasonable results in a sensible time scale. Exhaustive searches would take a prohibitive amount of time.

8.5 Processing Results

Due to the unique nature of the artificial olfactory mucosa, different approaches to displaying the data were considered. As less than 60% of the variance was contained in the first 2 principal components on average, the PCA results were plotted using the first 3 principal components, which contained 80%+ of the variance. The results from the neural net would however been presented in an illustrative format as well as the confusion matrix which characterises the network.

8.5.1 Principal Components Analysis

Figure 8.6 shows a selection of PCA results from the processing of the artificial olfactory mucosa.

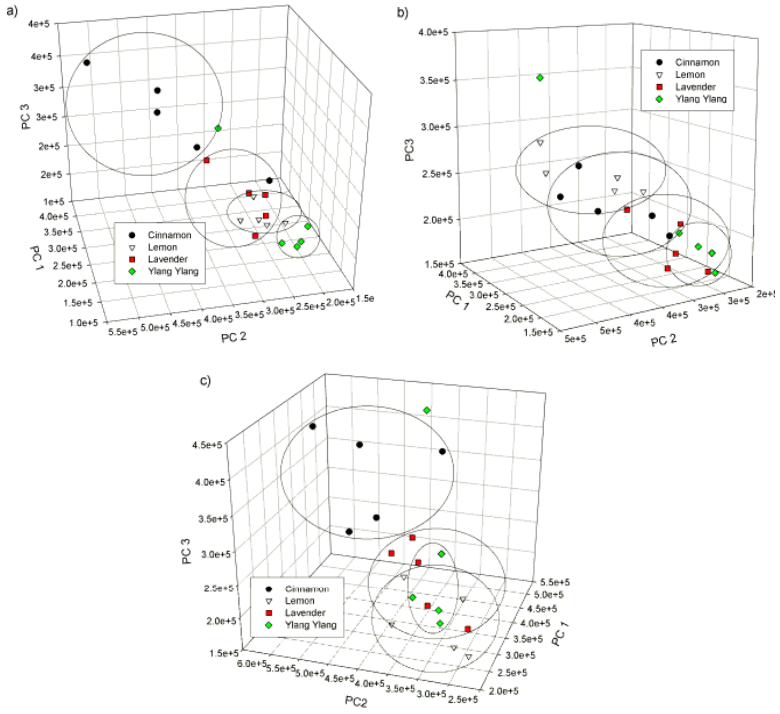


Figure 8.6 3-dimensional PCA plots resulting from processing. a) Front Array × Polar Array b) Front Array × Non-Polar Array c) Polar Array × Non-Polar Array

8.5.2 Probabilistic Neural Network

Figure 8.7 shows a graphical illustration of the peak magnitudes of the convolution characteristic signals obtained from the reduced set of functioning pairs of sensors. Table 8.2 summarises the accuracy of the PNN in classifying the testing vectors during the bootstrap training process.

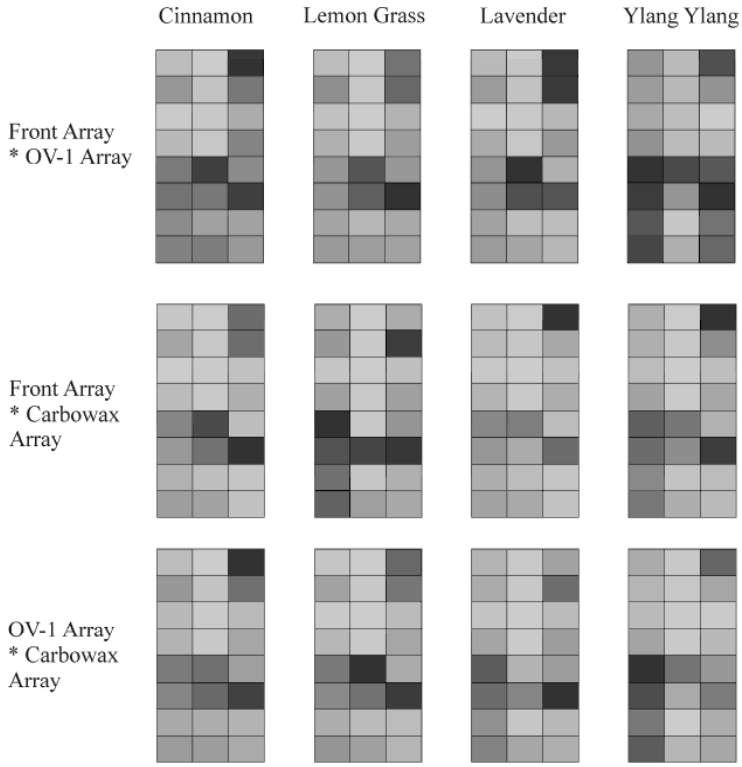


Figure 8.7 Graphical illustrations of the averaged peak magnitude of the convolution characteristic signal. A lighter shade of grey represents a higher relative peak magnitude. The signals are less correlated than the initial spatial data shown in Figure 8.3.

Table 8.2 Classification Accuracy of the PNN, between array pairs front * OV-1 arrays (F*O), front * carbowax arrays (F*C) and OV-1 and carbowax arrays (O*C)

Array Pair	Bootstrap Train and Test Classification Accuracy (%)					Average (%)
F*O	50	75	75	75	100	75
F*C	100	100	100	75	50	85
O*C	75	75	75	75	25	65

8.5.3 Analysis of Results

Comparing Figure 8.3 with Figure 8.6 or Figure 8.7 illustrates how processing affects the data and makes the differences between each analyte clearer. In Figure 8.3, while differences can be seen, the picture is unclear with only slight differences between each image, indicating a strong correlation between all the data. In comparison, the differences in Figure 8.7 are much clearer and more distinct, and it becomes easy to understand how the differences between the analytes could be identified.

For the e-mucosa device, a significant proportion of the sensors appear to operate poorly, yet classification schemes perform well after the convolution processing (perhaps similar to a feature of the human system with noisy olfactory receptors). The PCA results show good grouping of similar results, but the groups are quite close and difficult to separate using regression. The graphical approach shown in Figure 8.7 highlights the differences, and could form the basis of an effective matching classification scheme. The PNN results were disappointing, but with further refining of the neural network could form an effective classifier.

8.6 Summary

This chapter introduces an advanced electronic nose called the dual-channel e-mucosa, and considers how the convolution method can be applied to best effect. A scheme for applying the convolution method is proposed and used to characterise the system. Several novel display and classification techniques are also considered, presenting this complex new data in a clearer way.

This system poses several unique challenges. The unreliability of the sensors is overcome by both the redundant design of the system and censorship during the data processing stages. The data processing is successfully carried out and a different approach to classification is considered. An illustrative approach highlights how the processing affects the data and the features that can be used to identify different analytes.

The statistical and neural network approaches to classification under-perform when combined with this experimental technology. However, as both the technology and techniques are refined, there is clear scope to improve the results obtained from these methods.

Overall, the convolution method has been shown in a successful application to a very new technology under development, performing its purpose adequately. This complex and difficult data set produced sub-par results, but the classification was performed successfully.

8.7 References

[8.1] F. K. Che Harun, "Mimicking the Human Olfactory System: A Portable e-Mucosa", Ph.D. dissertation, School of Engineering, University of Warwick, Coventry, UK, 2009

[8.2] F. K. Che Harun, J. E. Taylor, J. A. Covington and J. W. Gardner, "An Electronic Nose Employing Dual-Channel Odour Separation Columns with Large Chemosensor Arrays for Advanced Odour Discrimination", *Sensor and Actuator B: Chemical*, vol. 141, Issue 1, pp. 134-140, 2009

[8.3] F.K.Che Harun, P.H. King, J.A. Covington and J.W. Gardner, "Novel Gas Chromatographic Microsystem With Very Large Sensor Arrays For Advanced Odour Discrimination", IEEE Sensors 2007. Presented on October 28 - 31, 2007 in Atlanta, USA.

[8.4] S. L. Tan, "Smart Chemical Sensing: Towards a Nose-on-a-Chip," Ph.D. dissertation, School of Engineering, University of Warwick, Coventry, UK, 2005

[8.5] S. Banks, *Signal Processing, Image Processing and Pattern Recognition*, London: Prentice Hall, 1990

[8.6] F-L. Luo and R. Unbehauen, *Applied Neural Networks for Signal Processing*, Cambridge: Cambridge University Press, 1997

[8.7] B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, 1996

Chapter 9: Conclusions

9.1 Overview

In this thesis, the changing field of processing for chemical sensing has been investigated. The advancement in our understanding of the olfactory system in nature has led to the realisation of the important role played by nasal chromatography across the olfactory epithelium. This new knowledge has in turn led to new devices in the field known as artificial olfaction, with the aim of producing artificial devices capable of distinguishing odour or airborne chemical analytes. Early artificial olfactory devices, called electronic noses, consisted of an array of sensors, similar to the olfactory cilia. These devices were well served by time-independent and linear processing techniques, a low-cost solution to the problem. The new knowledge of nasal chromatography has led to a new wave of artificial olfactory devices, collectively termed artificial olfactory mucosa devices. During the development of these devices, it was quickly realised that new processing approaches would be required to fully utilise the spatio-temporal data being produced by these new devices.

In response to this, time-dependent processing approaches were investigated. Already well established in other technical and data analysis fields, these methods were quickly applied to these new devices. However, these processes

were computationally intensive, and the devices being used were large, costly and slow. A drive for smaller, portable devices capable of rapid analysis revealed a need for a new approach.

This thesis puts forward a new processing approach, based around convolution. This mathematical technique, which underpins the common time-dependent processing approaches, is largely underlooked as a processing method. A method is proposed to use convolution to process chemical sensor data, as a method applicable to a large range of devices and problems.

In Chapter 4, a simulation of an electronic nose is performed, and the use of convolution as a processing methodology undergoes validation. Various simulations of real-world noise problems systematically altered the data. By applying a methodology utilising convolution during the processing stages, the data were successfully classified when the processed data were combined with a PCA regression classification scheme, and were shown to be an improvement over the simpler time-independent techniques. Note was taken that the convolution methods showed a marked weakness to drift-type noise in the data. This success validated convolution as a possible candidate for processing electronic nose data, and the work was continued.

In Chapters 5 & 6, the convolution method was applied to two real-world data sets. One data set presented a classification problem (Chapter 5) and the other presented a quantification problem (Chapter 6). With the general methodology validated by the previous simulation, during the course of these two processing

period, further investigation into the adaptability of the convolution method was taken. In Chapter 5, additional features of the convolution 'characteristic signal' were investigated, as well as the performance of the processed data when used with a non-parametric classification approach, in this case an artificial neural network. In Chapter 6, quantification presented a radically different problem to the classification carried out in the simulation. The different challenges and processing approach were attempted with the convolution method.

In processing these two sets of data, the convolution method once again validated itself as usable in these diverse circumstances, and showed that further refinement of method was possible. The method continued to successfully classify data correctly, whether used with parametric or non-parametric classifiers, and at a greater accuracy than these same methods without the convolution processing. In the quantification problem, it was shown that certain features of the convolution signal could be used to perform this task better than simpler alternatives, even though the final results had a significant margin of error.

Thus far however, the convolution method had only been applied to electronic nose data, data which is fully usable with time-independent statistical approaches. In Chapter 7, a simulation of an artificial olfactory mucosa is presented. This novel device generates the spatio-temporal data which requires time-dependent processing methods to be fully utilised. Tested alongside other signal-combining methods, the convolution method was shown to be superior over the other forms of processing, although maintaining the weakness to drift-noise shown in the first

simulation. However, the results show the superior performance of the AOM device over the e-nose device when subjected to the same simulation criteria.

In Chapter 8, the convolution method was applied to an advanced Dual-Channel Artificial Olfactory Mucosa. The experimental data from this very-large array device posed several unique challenges to the convolution data: a much larger data space, redundancy, spatial sensor correlation and 2 different retentive channels. Processing with the convolution method progressed, developing extensions to the techniques developed over the previous chapters. A graphical and non-parametric analysis approach proved to provide a solution to this tricky combination of factors, with the convolution method at the heart of the processing chain. It was shown that this technique could be adapted readily to this experimental device alongside novel methods of data presentation that are more suitable for this unique data space. The processing performed well on less than ideal data, producing an overall accuracy of 85%.

9.2 Future Work

Overall, this thesis has shown that the convolution method is a viable processing approach for the analysis of electronic nose, chemical sensing and artificial olfactory mucosa data. As more advanced and stable devices become available, the speed and ease of implementation of the convolution method, combined with good accuracy, should see the application of this method to rapid response devices and devices which require a good processing approach for multiple

applications. Future work should further integrate the convolution method in processing methodologies and refine the features extracted from the characteristic signal to produce even better results, or investigate entirely new features contained within this signal for use in future applications.

As the focus of this work was on the development and validation of a novel approach to processing, the data utilised during this process focused on analysis problems consisting of simple odours. Using these problems, which work reliably with current processing approaches, allowed a clear point of comparison from judging the value of the novel methods being developed. Now these processing methods have been validated with simple odours, further work can be carried out in presenting the convolution method more challenging problems, involving complex odours, taints and less ideal environments.

Appendix A: MATLAB Functions

In this section, several examples of the core MATLAB functions used to carry out the work presented in this thesis. These functions were created and used in MATLAB v7.4.0 (R2007a).

simulation_functions.m

This function is the core function of the simulations carried out in Chapters 4 and 7. In this function, characteristic signals are generated from input data, and the area of each characteristic signal is returned. The parameters determine the sensors, pre-processing and noise used when processing the signal, and how many repeat samples to perform for the simulation of these parameters.

```
function [c p d] =
simulation_functions(data1,data2,data3,a,b,ar,repeats,noise,nv)
%Compute convolution, product and difference functions of data
%Possible pre-processing raw,autoscaling,autoranging
%
%Input args: data sets, column A, column B, preprocess, repeats, noise type,
noise value
%Output args: area data sets of convolution, product and difference functions

%extract required data columns
E1 = data1(:,a);
E2 = data1(:,b);
T1 = data2(:,a);
T2 = data2(:,b);
M1 = data3(:,a);
M2 = data3(:,b);

%generate time and shift vectors from data
tE = data1(:,1);
tcE = [ flipud( -tE(2:length(tE)) ) ; tE];
tcE = tcE';

tT = data2(:,1);
tcT = [ flipud( -tT(2:length(tT)) ) ; tT];
tcT = tcT';

tM = data3(:,1);
tcM = [ flipud( -tM(2:length(tM)) ) ; tM];
tcM = tcM';

for i = 1:repeats
```

```
%add some noise (normal)

%sample varience
if (noise==1)
    n1 = 1+(nv*randn(1));
    n2 = 1+(nv*randn(1));
    A(1:length(E1),i) = E1 .* n1;
    B(1:length(E2),i) = E2 .* n2;
    n1 = 1+(nv*randn(1));
    n2 = 1+(nv*randn(1));
    A(1:length(T1),i+repeats) = T1 .* n1;
    B(1:length(T2),i+repeats) = T2 .* n2;
    n1 = 1+(nv*randn(1));
    n2 = 1+(nv*randn(1));
    A(1:length(M1),i+2*repeats) = M1 .* n1;
    B(1:length(M2),i+2*repeats) = M2 .* n2;
end

%SNR
if (noise==2)
    n1 = max(abs(E1))*nv*randn(length(E1),1);
    n2 = max(abs(E2))*nv*randn(length(E2),1);
    A(1:length(E1),i) = E1 + n1;
    B(1:length(E2),i) = E2 + n2;
    n1 = max(abs(T1))*nv*randn(length(T1),1);
    n2 = max(abs(T2))*nv*randn(length(T2),1);
    A(1:length(T1),i+repeats) = T1 + n1;
    B(1:length(T2),i+repeats) = T2 + n2;
    n1 = max(abs(M1))*nv*randn(length(M1),1);
    n2 = max(abs(M2))*nv*randn(length(M2),1);
    A(1:length(M1),i+2*repeats) = M1 + n1;
    B(1:length(M2),i+2*repeats) = M2 + n2;
end

%baseline drift
if (noise==3)
    n = nv*randn(1);
    n1 = max(abs(E1))*n;
    n2 = max(abs(E2))*n;
    A(1:length(E1),i) = E1 + n1;
    B(1:length(E2),i) = E2 + n2;
    n = nv*randn(1);
    n1 = max(abs(T1))*n;
    n2 = max(abs(T2))*n;
    A(1:length(T1),i+repeats) = T1 + n1;
    B(1:length(T2),i+repeats) = T2 + n2;
    n = nv*randn(1);
    n1 = max(abs(M1))*n;
    n2 = max(abs(M2))*n;
    A(1:length(M1),i+2*repeats) = M1 + n1;
    B(1:length(M2),i+2*repeats) = M2 + n2;
end

%sample varience + SNR
if (noise==4)
    maxE1 = max(abs(E1));
    maxE2 = max(abs(E2));
    maxT1 = max(abs(T1));
    maxT2 = max(abs(T2));
    maxM1 = max(abs(M1));
    maxM2 = max(abs(M2));

    n1 = 1+(nv*randn(1));
    n2 = 1+(nv*randn(1));
    A(1:length(E1),i) = E1 .* n1;
    B(1:length(E2),i) = E2 .* n2;
    n1 = 1+(nv*randn(1));
    n2 = 1+(nv*randn(1));
    A(1:length(T1),i+repeats) = T1 .* n1;
    B(1:length(T2),i+repeats) = T2 .* n2;
    n1 = 1+(nv*randn(1));
    n2 = 1+(nv*randn(1));
    A(1:length(M1),i+2*repeats) = M1 .* n1;
    B(1:length(M2),i+2*repeats) = M2 .* n2;
```

```
        n1 = maxE1*nv*randn(length(E1),1);
        n2 = maxE2*nv*randn(length(E2),1);
        A(1:length(E1),i) = E1 + n1;
        B(1:length(E2),i) = E2 + n2;
        n1 = maxT1*nv*randn(length(T1),1);
        n2 = maxT2*nv*randn(length(T2),1);
        A(1:length(T1),i+repeats) = T1 + n1;
        B(1:length(T2),i+repeats) = T2 + n2;
        n1 = maxM1*nv*randn(length(M1),1);
        n2 = maxM2*nv*randn(length(M2),1);
        A(1:length(M1),i+2*repeats) = M1 + n1;
        B(1:length(M2),i+2*repeats) = M2 + n2;
    end

    %sample varience + baseline drift
    if (noise==5)
        maxE1 = max(abs(E1));
        maxE2 = max(abs(E2));
        maxT1 = max(abs(T1));
        maxT2 = max(abs(T2));
        maxM1 = max(abs(M1));
        maxM2 = max(abs(M2));

        n1 = 1+(nv*randn(1));
        n2 = 1+(nv*randn(1));
        A(1:length(E1),i) = E1 .* n1;
        B(1:length(E2),i) = E2 .* n2;
        n1 = 1+(nv*randn(1));
        n2 = 1+(nv*randn(1));
        A(1:length(T1),i+repeats) = T1 .* n1;
        B(1:length(T2),i+repeats) = T2 .* n2;
        n1 = 1+(nv*randn(1));
        n2 = 1+(nv*randn(1));
        A(1:length(M1),i+2*repeats) = M1 .* n1;
        B(1:length(M2),i+2*repeats) = M2 .* n2;

        n = nv*randn(1);
        n1 = maxE1*n;
        n2 = maxE2*n;
        A(1:length(E1),i) = E1 + n1;
        B(1:length(E2),i) = E2 + n2;
        n = nv*randn(1);
        n1 = maxT1*n;
        n2 = maxT2*n;
        A(1:length(T1),i+repeats) = T1 + n1;
        B(1:length(T2),i+repeats) = T2 + n2;
        n = nv*randn(1);
        n1 = maxM1*n;
        n2 = maxM2*n;
        A(1:length(M1),i+2*repeats) = M1 + n1;
        B(1:length(M2),i+2*repeats) = M2 + n2;
    end

    %SNR + baseline drift
    if (noise==6)
        maxE1 = max(abs(E1));
        maxE2 = max(abs(E2));
        maxT1 = max(abs(T1));
        maxT2 = max(abs(T2));
        maxM1 = max(abs(M1));
        maxM2 = max(abs(M2));

        n1 = maxE1*nv*randn(length(E1),1);
        n2 = maxE2*nv*randn(length(E2),1);
        A(1:length(E1),i) = E1 + n1;
        B(1:length(E2),i) = E2 + n2;
        n1 = maxT1*nv*randn(length(T1),1);
        n2 = maxT2*nv*randn(length(T2),1);
        A(1:length(T1),i+repeats) = T1 + n1;
        B(1:length(T2),i+repeats) = T2 + n2;
        n1 = maxM1*nv*randn(length(M1),1);
        n2 = maxM2*nv*randn(length(M2),1);
        A(1:length(M1),i+2*repeats) = M1 + n1;
        B(1:length(M2),i+2*repeats) = M2 + n2;

        n = nv*randn(1);
```

```
        n1 = maxE1*n;
        n2 = maxE2*n;
        A(1:length(E1),i) = E1 + n1;
        B(1:length(E2),i) = E2 + n2;
        n = nv*randn(1);
        n1 = maxT1*n;
        n2 = maxT2*n;
        A(1:length(T1),i+repeats) = T1 + n1;
        B(1:length(T2),i+repeats) = T2 + n2;
        n = nv*randn(1);
        n1 = maxM1*n;
        n2 = maxM2*n;
        A(1:length(M1),i+2*repeats) = M1 + n1;
        B(1:length(M2),i+2*repeats) = M2 + n2;
    end

end

%autorange the data, if desired
if (ar == 1)
    mxA = max(max(A));
    mxB = max(max(B));
    mnA = min(min(A));
    mnB = min(min(B));
    arA = mxA - mnA;
    arB = mxB - mnB;

    for i=1:3*repeats
        A(:,i) = A(:,i) / arA;
        B(:,i) = B(:,i) / arB;
    end
end

%autoscale the data, if desired
if (ar == 2)
    mxA = max(max(abs(A)));
    mxB = max(max(abs(B)));

    for i=1:3*repeats
        A(:,i) = abs(A(:,i)) / mxA;
        B(:,i) = abs(B(:,i)) / mxB;
    end
end

%find convolution area, product area and difference area
for i=1:repeats

    c(i) = trapz(tcE,conv(A(1:length(tE),i),B(1:length(tE),i)));
    c(i+repeats) =
trapz(tcT,conv(A(1:length(tT),i+repeats),B(1:length(tT),i+repeats)));
    c(i+2*repeats) =
trapz(tcM,conv(A(1:length(tM),i+2*repeats),B(1:length(tM),i+2*repeats)));

    p(i) = trapz(tE,sign(A(1:length(tE),i) .* B(1:length(tE),i)) .*
sqrt(abs(A(1:length(tE),i)) .* abs(B(1:length(tE),i))));
    p(i+repeats) = trapz(tT,sign(A(1:length(tT),i+repeats) .*
B(1:length(tT),i+repeats)) .* sqrt(abs(A(1:length(tT),i+repeats)) .*
abs(B(1:length(tT),i+repeats))));
    p(i+2*repeats) = trapz(tM,sign(A(1:length(tM),i+2*repeats) .*
B(1:length(tM),i+2*repeats)) .* sqrt(abs(A(1:length(tM),i+2*repeats)) .*
abs(B(1:length(tM),i+2*repeats))));

    d(i) = trapz(tE,A(1:length(tE),i) - B(1:length(tE),i));
    d(i+repeats) = trapz(tT,A(1:length(tT),i+repeats) -
B(1:length(tT),i+repeats));
    d(i+2*repeats) = trapz(tM,A(1:length(tM),i+2*repeats) -
B(1:length(tM),i+2*repeats));

end
```

The following is an example of a MATLAB script making use of this function:

```
%load and enumerate data
load_data

%Noise type 1

repeats=20;
n=1;
nv=0.05;

randn('state',sum(100*clock)); %reseed RNG

%autoranged nosiy data set
[C(:,1) P(:,1) D(:,1)] =
simulation_functions(EtnC,TolC,MixC,S1A,S1B,1,repeats,n,nv);
[C(:,2) P(:,2) D(:,2)] =
simulation_functions(EtnC,TolC,MixC,S2A,S2B,1,repeats,n,nv);
[C(:,3) P(:,3) D(:,3)] =
simulation_functions(EtnC,TolC,MixC,S3A,S3B,1,repeats,n,nv);
[C(:,4) P(:,4) D(:,4)] =
simulation_functions(EtnC,TolC,MixC,S4A,S4B,1,repeats,n,nv);
[C(:,5) P(:,5) D(:,5)] =
simulation_functions(EtnC,TolC,MixC,S5A,S5B,1,repeats,n,nv);
[C(:,6) P(:,6) D(:,6)] =
simulation_functions(EtnC,TolC,MixC,S6A,S6B,1,repeats,n,nv);
[C(:,7) P(:,7) D(:,7)] =
simulation_functions(EtnC,TolC,MixC,S7A,S7B,1,repeats,n,nv);
[C(:,8) P(:,8) D(:,8)] =
simulation_functions(EtnC,TolC,MixC,S8A,S8B,1,repeats,n,nv);
[C(:,9) P(:,9) D(:,9)] =
simulation_functions(EtnC,TolC,MixC,S9A,S9B,1,repeats,n,nv);
[C(:,10) P(:,10) D(:,10)] =
simulation_functions(EtnC,TolC,MixC,S10A,S10B,1,repeats,n,nv);

save conv_ar_sv.txt C -ascii
save prod_ar_sv.txt P -ascii
save diff_ar_sv.txt D -ascii

C = [];
P = [];
D = [];

%raw data set
[C(:,1) P(:,1) D(:,1)] =
simulation_functions(EtnC,TolC,MixC,S1A,S1B,0,repeats,n,nv);
[C(:,2) P(:,2) D(:,2)] =
simulation_functions(EtnC,TolC,MixC,S2A,S2B,0,repeats,n,nv);
[C(:,3) P(:,3) D(:,3)] =
simulation_functions(EtnC,TolC,MixC,S3A,S3B,0,repeats,n,nv);
[C(:,4) P(:,4) D(:,4)] =
simulation_functions(EtnC,TolC,MixC,S4A,S4B,0,repeats,n,nv);
[C(:,5) P(:,5) D(:,5)] =
simulation_functions(EtnC,TolC,MixC,S5A,S5B,0,repeats,n,nv);
[C(:,6) P(:,6) D(:,6)] =
simulation_functions(EtnC,TolC,MixC,S6A,S6B,0,repeats,n,nv);
[C(:,7) P(:,7) D(:,7)] =
simulation_functions(EtnC,TolC,MixC,S7A,S7B,0,repeats,n,nv);
[C(:,8) P(:,8) D(:,8)] =
simulation_functions(EtnC,TolC,MixC,S8A,S8B,0,repeats,n,nv);
[C(:,9) P(:,9) D(:,9)] =
simulation_functions(EtnC,TolC,MixC,S9A,S9B,0,repeats,n,nv);
[C(:,10) P(:,10) D(:,10)] =
simulation_functions(EtnC,TolC,MixC,S10A,S10B,0,repeats,n,nv);

save conv_raw_sv.txt C -ascii
save prod_raw_sv.txt P -ascii
save diff_raw_sv.txt D -ascii

C = [];
P = [];
D = [];

%autoscaled nosiy data set
```

```
[C(:,1) P(:,1) D(:,1)] =
simulation_functions(EtnC,TolC,MixC,S1A,S1B,2,repeats,n,nv);
[C(:,2) P(:,2) D(:,2)] =
simulation_functions(EtnC,TolC,MixC,S2A,S2B,2,repeats,n,nv);
[C(:,3) P(:,3) D(:,3)] =
simulation_functions(EtnC,TolC,MixC,S3A,S3B,2,repeats,n,nv);
[C(:,4) P(:,4) D(:,4)] =
simulation_functions(EtnC,TolC,MixC,S4A,S4B,2,repeats,n,nv);
[C(:,5) P(:,5) D(:,5)] =
simulation_functions(EtnC,TolC,MixC,S5A,S5B,2,repeats,n,nv);
[C(:,6) P(:,6) D(:,6)] =
simulation_functions(EtnC,TolC,MixC,S6A,S6B,2,repeats,n,nv);
[C(:,7) P(:,7) D(:,7)] =
simulation_functions(EtnC,TolC,MixC,S7A,S7B,2,repeats,n,nv);
[C(:,8) P(:,8) D(:,8)] =
simulation_functions(EtnC,TolC,MixC,S8A,S8B,2,repeats,n,nv);
[C(:,9) P(:,9) D(:,9)] =
simulation_functions(EtnC,TolC,MixC,S9A,S9B,2,repeats,n,nv);
[C(:,10) P(:,10) D(:,10)] =
simulation_functions(EtnC,TolC,MixC,S10A,S10B,2,repeats,n,nv);

save conv_as_sv.txt C -ascii
save prod_as_sv.txt P -ascii
save diff_as_sv.txt D -ascii

C = [];
P = [];
D = [];
```

area_functions.m

This is an adaptation of *simulation_functions.m* suitable for use with real data imported into MATLAB, and used in Chapters 5 and 8. The simulation noise and repeating processes have been removed. Additionally, the function also returns peak magnitude of the characteristic signals, and the characteristic signals themselves. Later versions would remove the product and difference signals to simplify the function once it was determined that convolution was the most useful characteristic signal.

```
function [c p d hc hp hd cs ps ds] =
area_functions(data1,data2,time1,time2,a,b,ar)
%Compute convolution, product and difference functions of data
%Possible pre-processing raw=0, autoranging=1, autoscaling=2
%
%Input args: data set A, data set B, time series A, time series B, sensor A,
sensor B, preprocessing
%Output args: area data sets, magnitude data sets, characteristic signals

%extract required data columns
A = data1(:,a);
B = data2(:,b);

tA = time1(:,a);
tB = time2(:,b);

%generate convolution time series
```

```
tcA = [ flipud( -tB(2:length(tB)) ) ; tA(1:length(tA))];

%autorange the data, if desired
if (ar == 1)
    mxA = max(max(A));
    mxB = max(max(B));
    mnA = min(min(A));
    mnB = min(min(B));
    arA = mxA - mnA;
    arB = mxB - mnB;
    A = A / arA;
    B = B / arB;
end

%autoscale the data, if desired
if (ar == 2)
    mxA = max(max(A));
    mxB = max(max(B));
    mnA = min(min(A));
    mnB = min(min(B));
    arA = mxA - mnA;
    arB = mxB - mnB;
    A = A - mnA;
    B = B - mnB;
    A = A / arA;
    B = B / arB;

    %mxA = max(max(abs(A)));
    %mxB = max(max(abs(B)));

    %A = abs(A) / mxA;
    %B = abs(B) / mxB;

end

%find convolution characteristic signal, difference characteristic signal and
product characteristic signal
cs = conv(A,B);
ps = sign(A .* B .* sqrt(abs(A)) .* abs(B));
ds = A-B;

%find area of each characteristic signal
c = trapz(tcA,cs);
p = trapz(tA,sign(A .* B .* sqrt(abs(A)) .* abs(B)));
d = trapz(tA,(A-B));

%find maximum magnitude of each characteristic signal
hc = max(cs);
hp = max(ps);
hd = max(ds);
```

The following is an example MATLAB script making use of the above function:

```
%load and enumerate data
load_data;

%pw1
[C(:,1) P(:,1) D(:,1)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S1A,S1B,1);
[C(:,2) P(:,2) D(:,2)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S2A,S2B,1);
[C(:,3) P(:,3) D(:,3)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S3A,S3B,1);
[C(:,4) P(:,4) D(:,4)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S4A,S4B,1);
[C(:,5) P(:,5) D(:,5)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S5A,S5B,1);
[C(:,6) P(:,6) D(:,6)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S6A,S6B,1);
[C(:,7) P(:,7) D(:,7)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S7A,S7B,1);
[C(:,8) P(:,8) D(:,8)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S8A,S8B,1);
[C(:,9) P(:,9) D(:,9)] = area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S9A,S9B,1);
[C(:,10) P(:,10) D(:,10)] =
area_functions('p3ct_etn_pw1','p3ct_tol_pw1',S10A,S10B,1);

save conv_pw1.txt C -ascii
save prod_pw1.txt P -ascii
save diff_pw1.txt D -ascii
```

```
C = [];
P = [];
D = [];

%pw5
[C(:,1) P(:,1) D(:,1)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S1A,S1B,1);
[C(:,2) P(:,2) D(:,2)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S2A,S2B,1);
[C(:,3) P(:,3) D(:,3)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S3A,S3B,1);
[C(:,4) P(:,4) D(:,4)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S4A,S4B,1);
[C(:,5) P(:,5) D(:,5)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S5A,S5B,1);
[C(:,6) P(:,6) D(:,6)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S6A,S6B,1);
[C(:,7) P(:,7) D(:,7)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S7A,S7B,1);
[C(:,8) P(:,8) D(:,8)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S8A,S8B,1);
[C(:,9) P(:,9) D(:,9)] = area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S9A,S9B,1);
[C(:,10) P(:,10) D(:,10)] =
area_functions('p3ct_etn_pw5','p3ct_tol_pw5',S10A,S10B,1);

save conv_pw5.txt C -ascii
save prod_pw5.txt P -ascii
save diff_pw5.txt D -ascii

C = [];
P = [];
D = [];
```

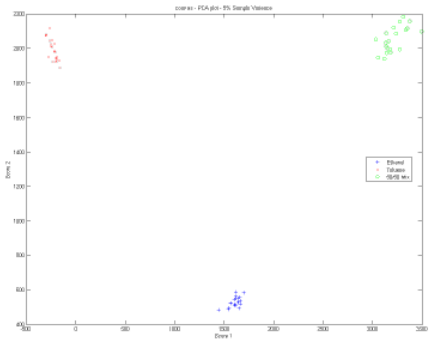
do_pls.m

This function is an adaptation of the PLS function obtained from http://www.mathworks.com/matlabcentral/fileexchange/18760-partial-least-squares-and-discriminant-analysis by Y. Cao, modified to be suitable for the data used in Chapter 6.

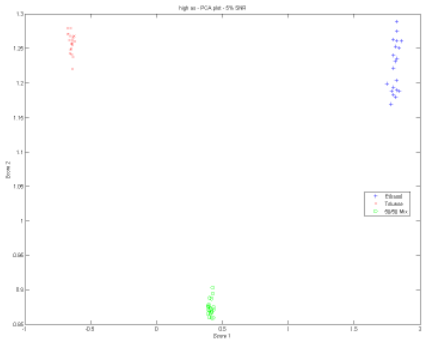
```
function [C] = do_pls(input)
data=load([input '.txt']);

% Four classes data, each has 8 samples and 32 variables.
x2=data(9:16,:);
x3=data(17:24,:);
x4=data(25:32,:);
x5=data(33:40,:);

% Combine training data with normalization
X = [x2;x3;x4;x5];
% Define class indicator as Y
Y = kron(eye(4),ones(8,1));
% Normalization
xmean = mean(X);
xstd = std(X);
ymean = mean(Y);
ystd = std(Y);
X = (X - xmean(ones(32,1),:))./xstd(ones(32,1),:);
Y = (Y - ymean(ones(32,1),:))./ystd(ones(32,1),:);
% Perform PLS
tol = 0.1 * 8 * 4;
[T,P,U,Q,B] = pls(X,Y,tol);
% Results
fprintf('Number of components retained: %i\n',size(B,1))

A(1:8,:) = x2 * (P*B*Q');
A(9:16,:) = x3 * (P*B*Q');
A(17:24,:) = x4 * (P*B*Q');
A(25:32,:) = x5 * (P*B*Q');
C = A .* ystd(ones(32,1),:) + ymean(ones(32,1),:);
save([input '_pls.txt'],'C','-ascii');
```

Appendix B: e-Nose Simulation PCA Results

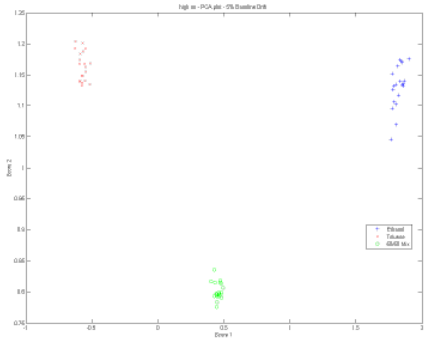
Autoscaling Pre-Processing



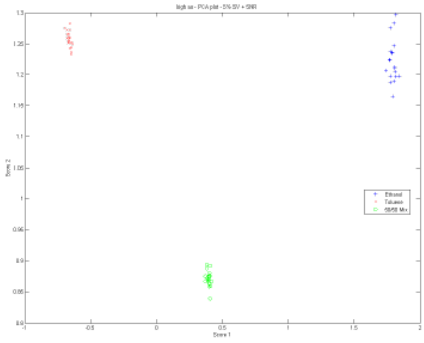
Peak Magnitude, Sample Variance



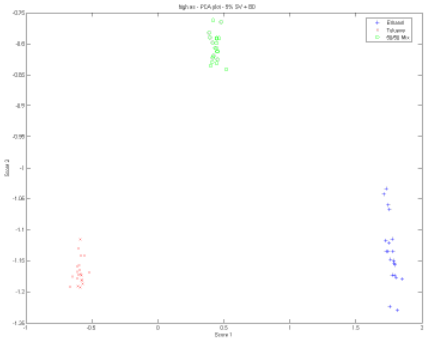
Peak Magnitude, Additive Noise



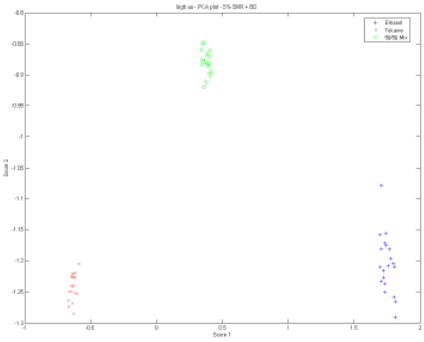
Peak Magnitude, Baseline Drift



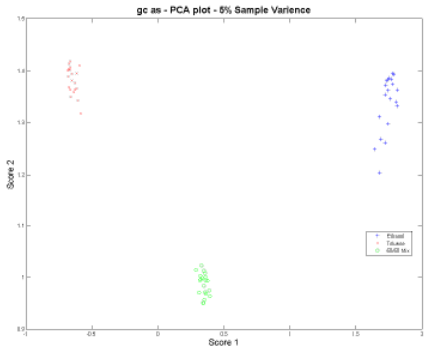
Peak Magnitude, SV + SNR



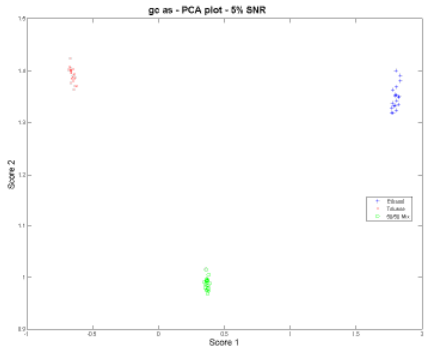
Peak Magnitude, SV + BD



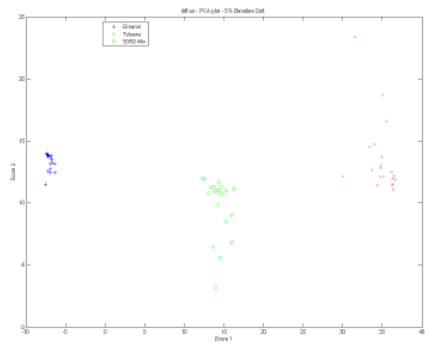
Peak Magnitude, SNR + BD



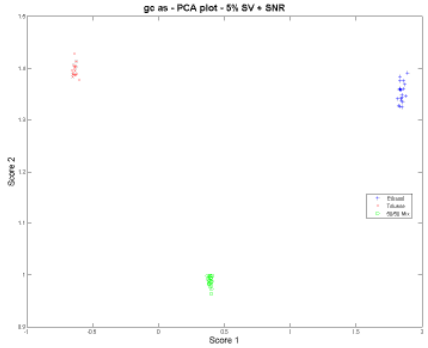
c-Nose Peak Magnitude, SV



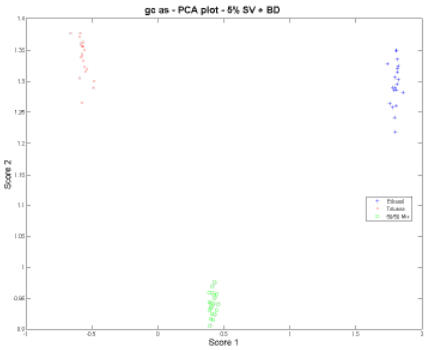
c-Nose Peak Magnitude, SNR



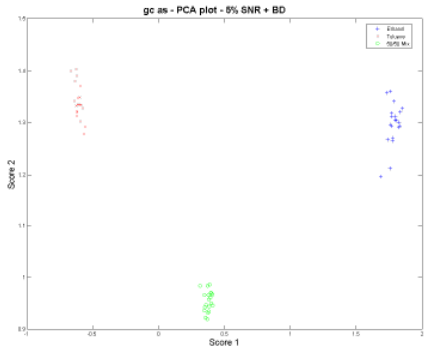
c-Nose Peak Magnitude, BD



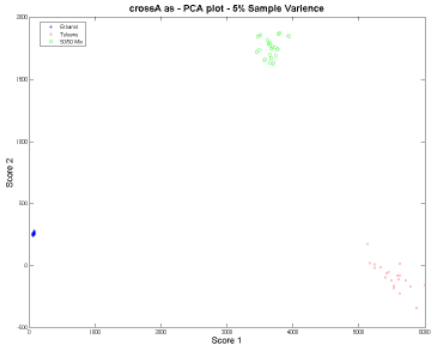
c-Nose Peak Magnitude, SV + SNR



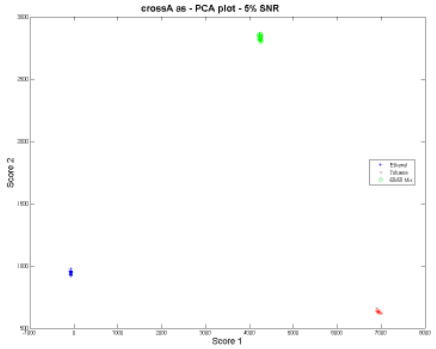
c-Nose Peak Magnitude, SV + BD



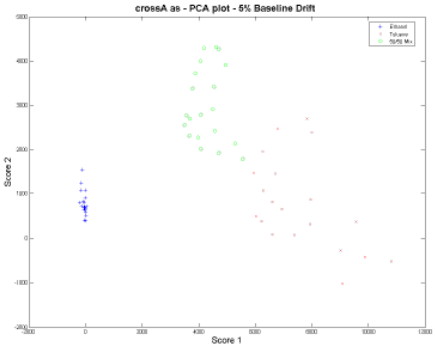
c-Nose Peak Magnitude, SNR + BD



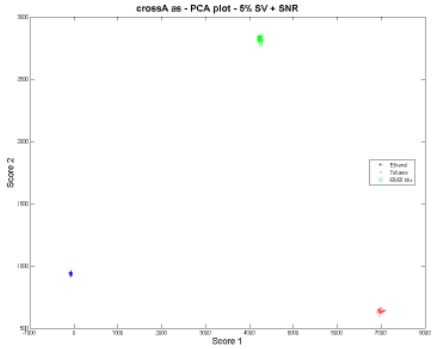
Cross-convolution, SV



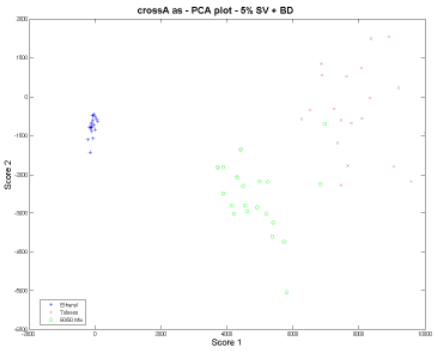
Cross-convolution, SNR



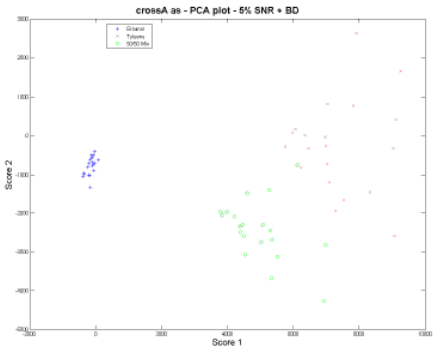
Cross-convolution, BD



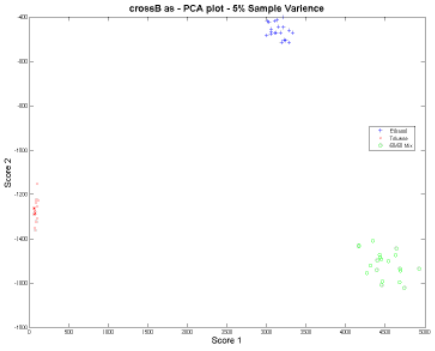
Cross-convolution, SV + SNR



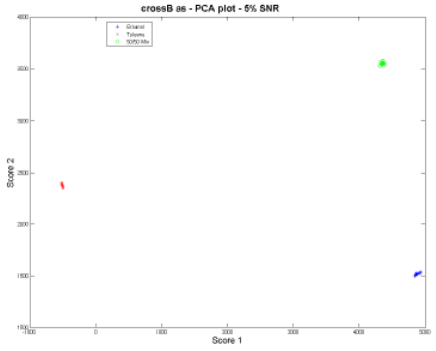
Cross-convolution, SV + BD



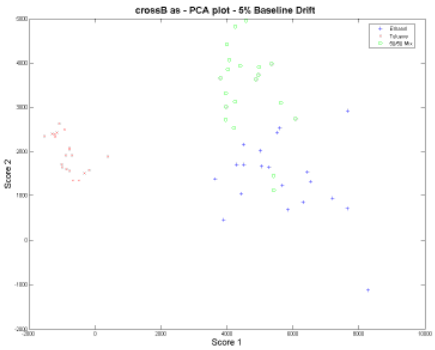
Cross-convolution, SNR + BD



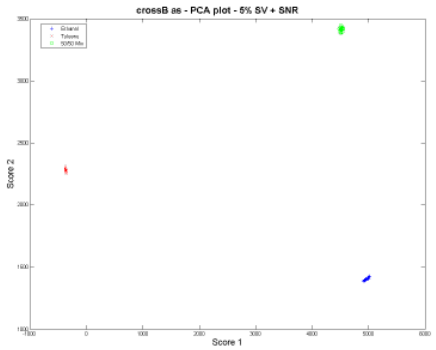
c-Nose Cross-convolution, SV



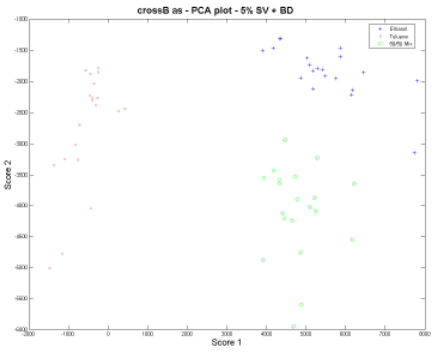
c-Nose Cross-convolution, SNR



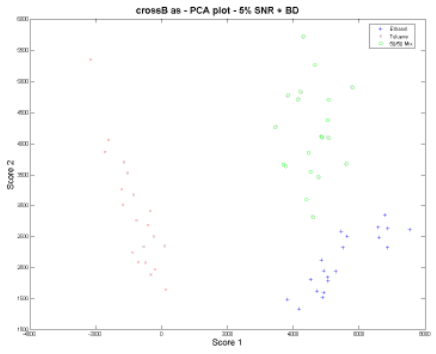
c-Nose Cross-convolution, BD



c-Nose Cross-convolution, SV + SNR

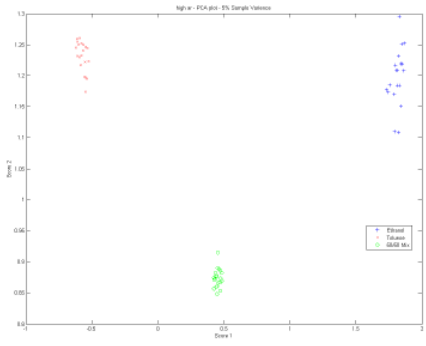


c-Nose Cross-convolution, SV + BD

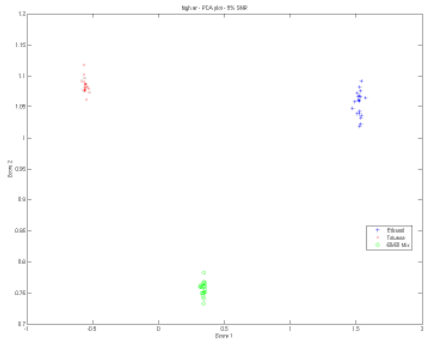


c-Nose Cross-convolution, SNR + BD

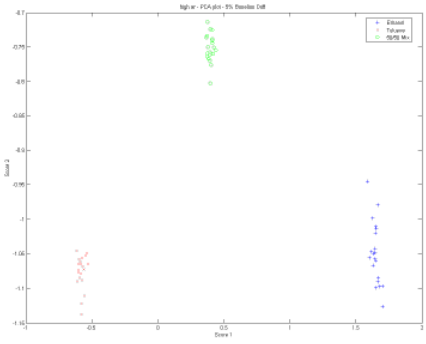
Autoranging Pre-processing



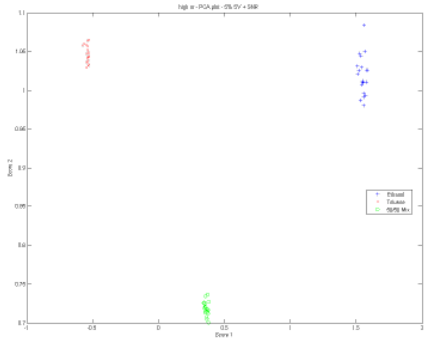
Peak Magnitude, SV



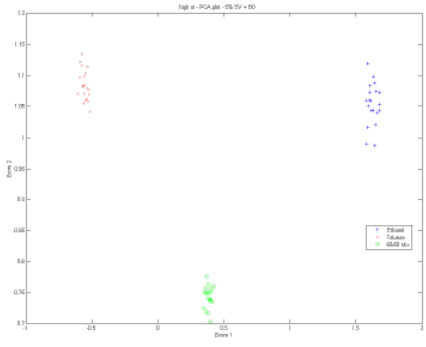
Peak Magnitude, SNR



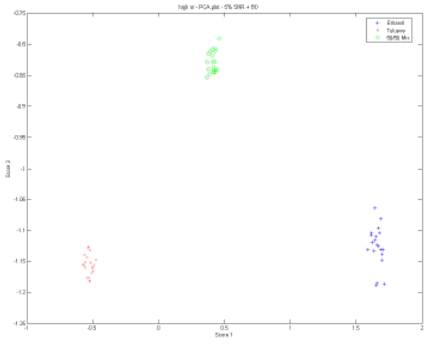
Peak Magnitude, BD



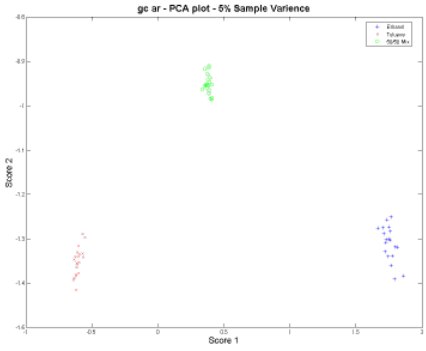
Peak Magnitude, SV + SNR



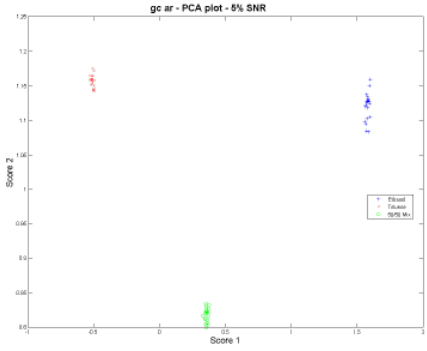
Peak Magnitude, SV + BD



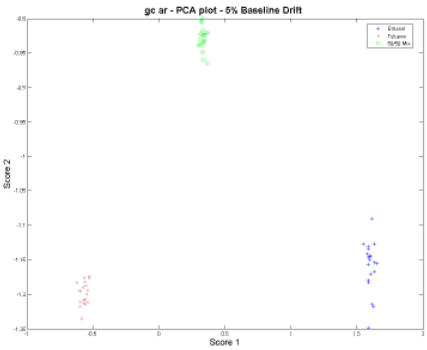
Peak Magnitude, SNR + BD



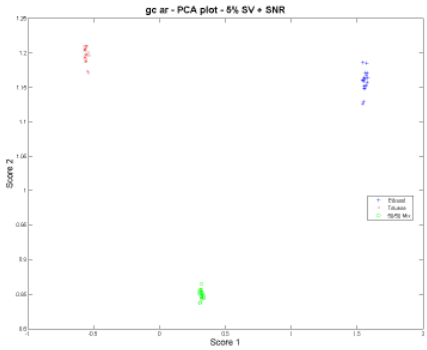
c-Nose Peak Magnitude, SV



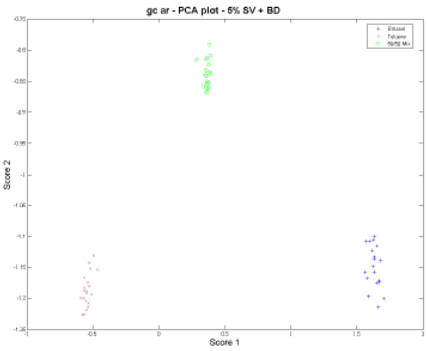
c-Nose Peak Magnitude, SNR



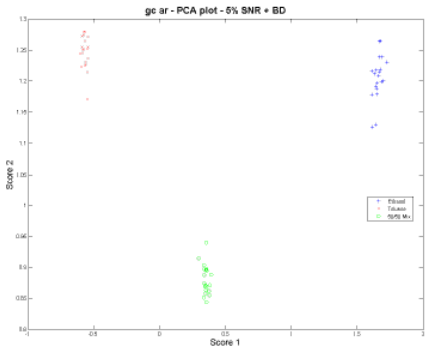
c-Nose Peak Magnitude, BD



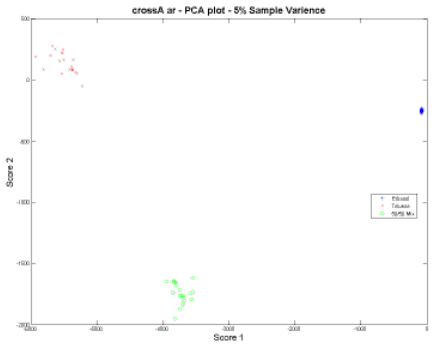
c-Nose Peak Magnitude, SV + SNR



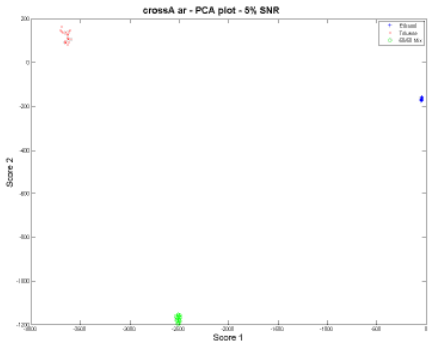
c-Nose Peak Magntide, SV + BD



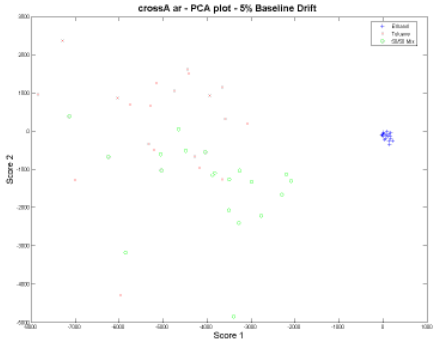
c-Nose Peak Magnitude, SNR + BD



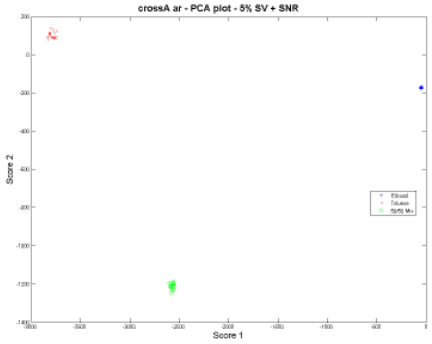
Cross-convolution, SV



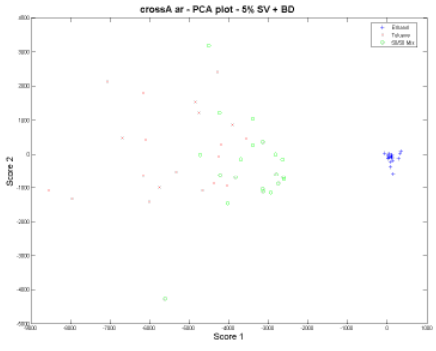
Cross-convolution, SNR



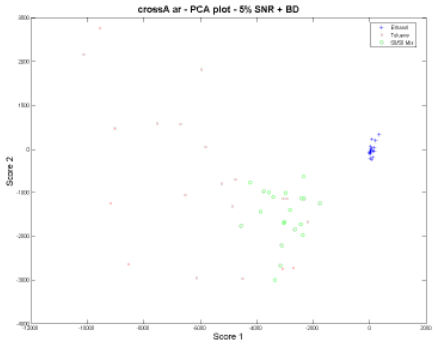
Cross-convolution, BD



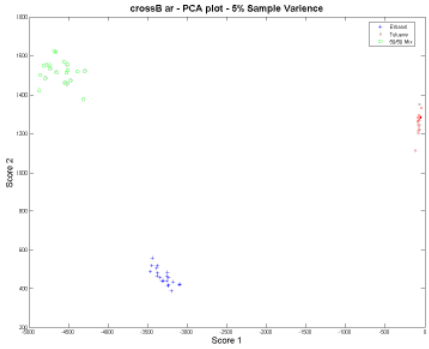
Cross-convolution, SV + SNR



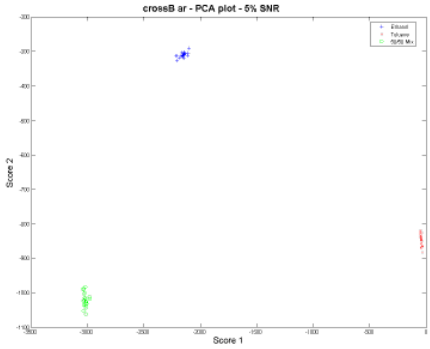
Cross-convolution, SV + BD



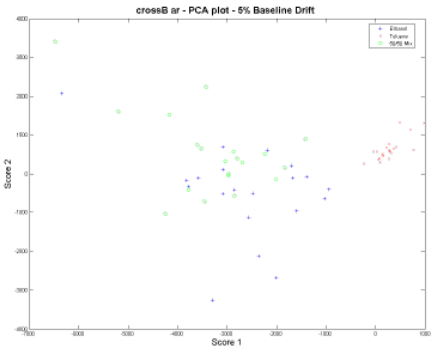
Cross-convolution, SNR + BD



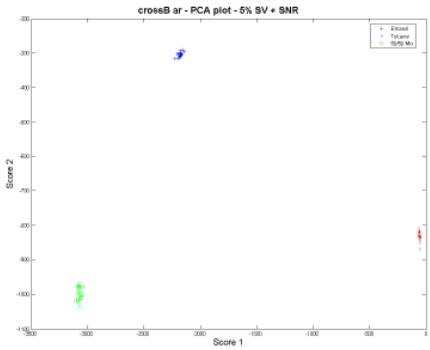
c-Nose Cross-convolution, SV



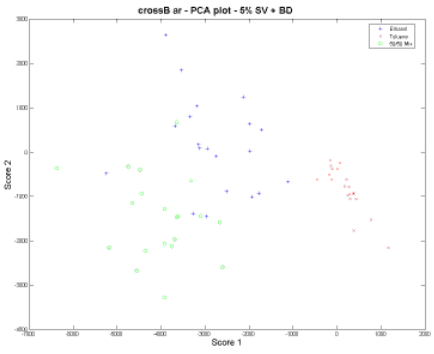
c-Nose Cross-convolution, SNR



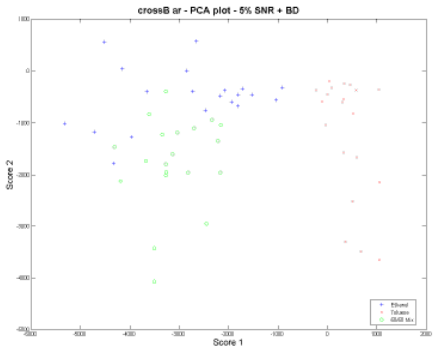
c-Nose Cross-convolution, BD



c-Nose Cross-convolution, SV + SNR

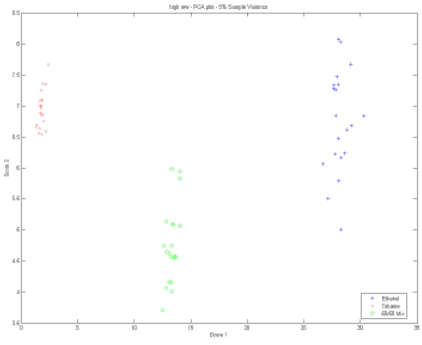


c-Nose Cross-convolution, SV + BD

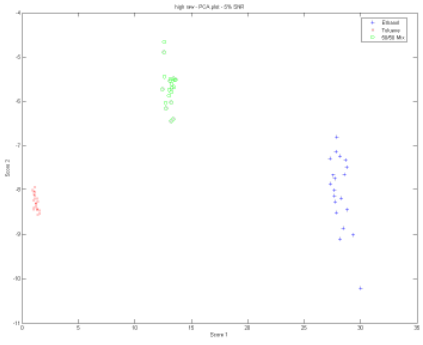


c-Nose Cross-convolution, SNR + BD

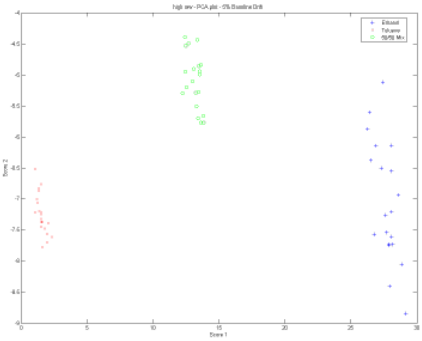
No Pre-processing



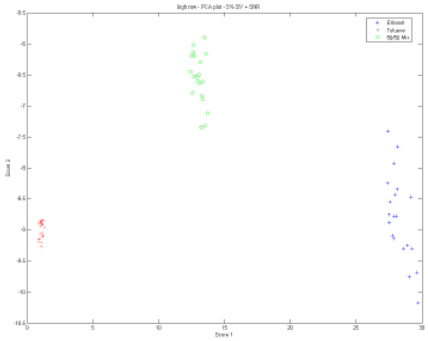
Peak Magnitude, SV



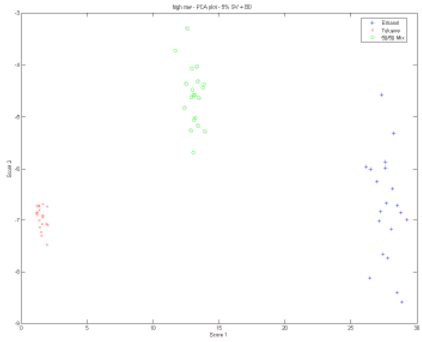
Peak Magnitude, SNR



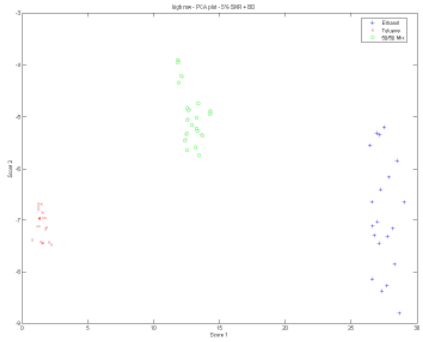
Peak Magnitude, BD



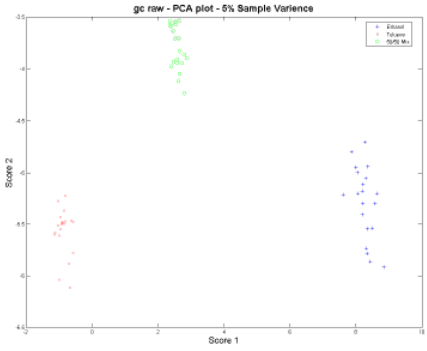
Peak Magnitude, SV + SNR



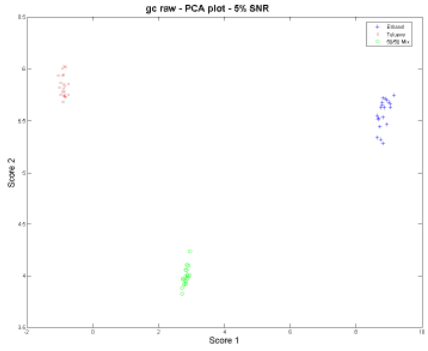
Peak Magnitude, SV + BD



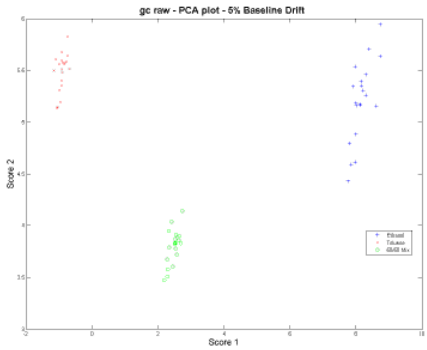
Peak Magnitude, SNR + BD



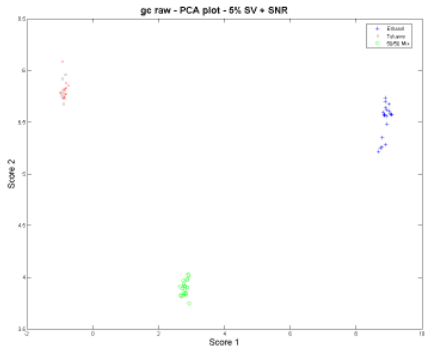
c-Nose Peak Magnitude, SV



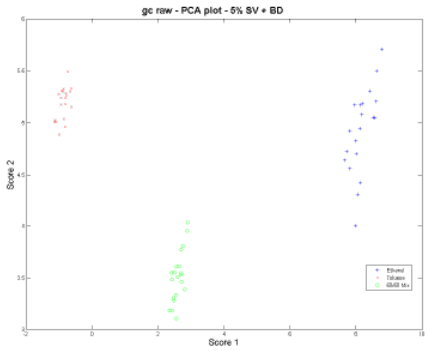
c-Nose Peak Magnitude, SNR



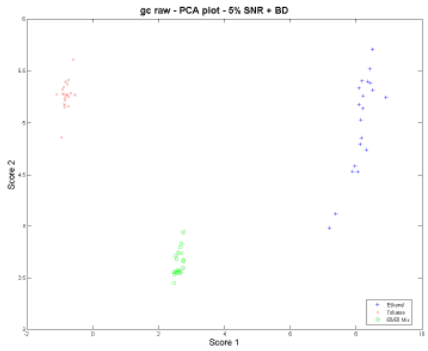
c-Nose Peak Magnitude, BD



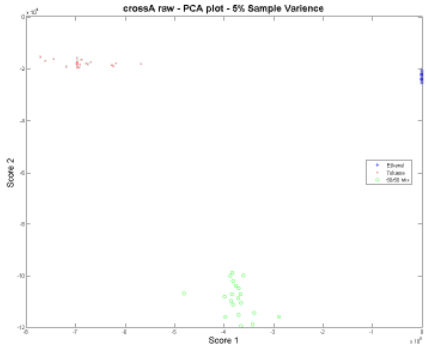
c-Nose Peak Magnitude, SV + SNR



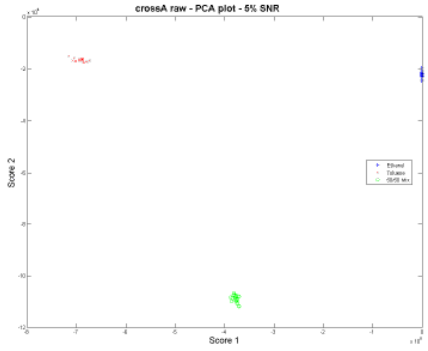
c-Nose Peak Magnitude, SV + BD



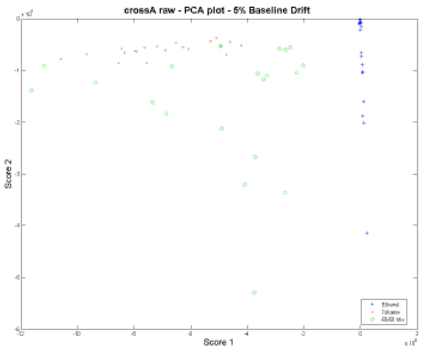
c-Nose Peak Magnitude, SNR + BD



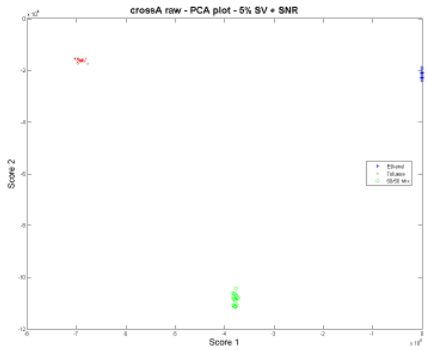
Cross-convolution, SV



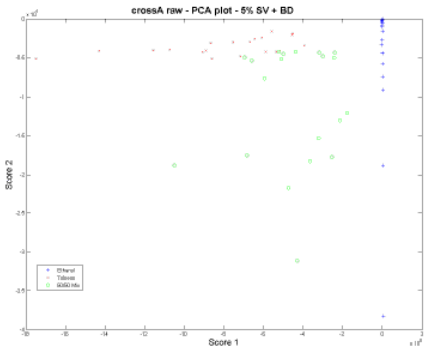
Cross-convolution, SNR



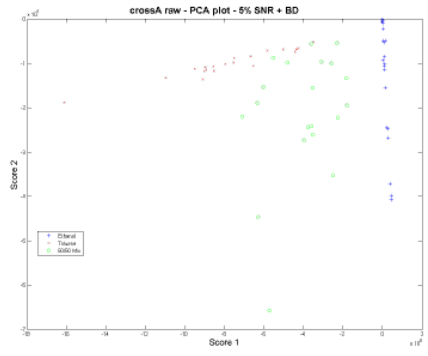
Cross-convolution, BD



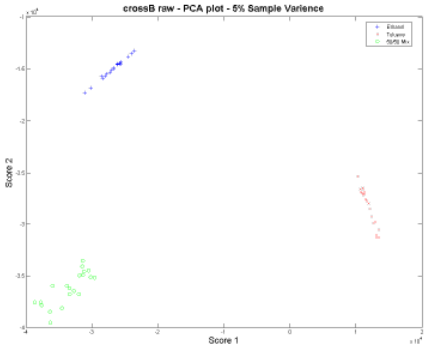
Cross-convolution, SV + SNR



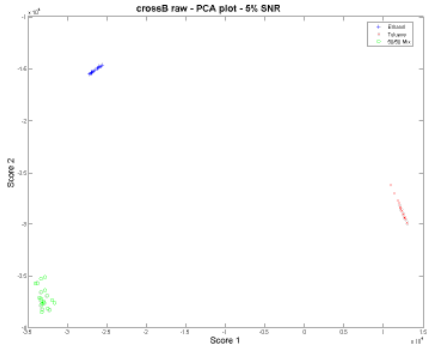
Cross-convolution, SV + BD



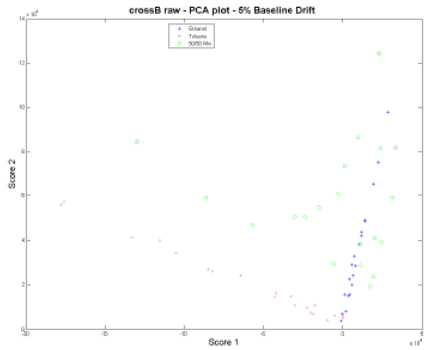
Cross-convolution, SNR + BD



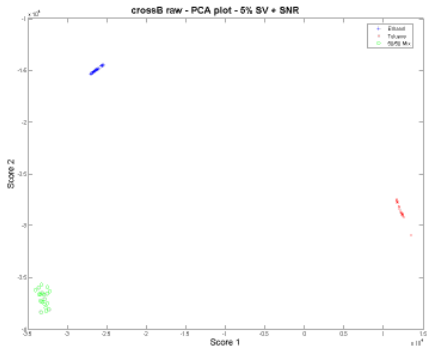
c-Nose Cross-convolution, SV



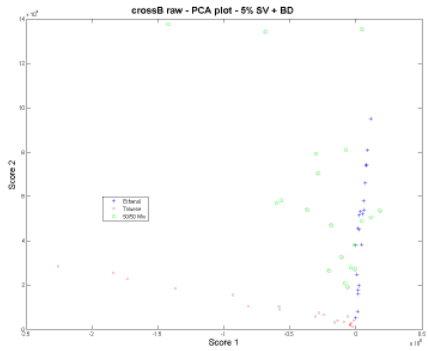
c-Nose Cross-convolution, SNR



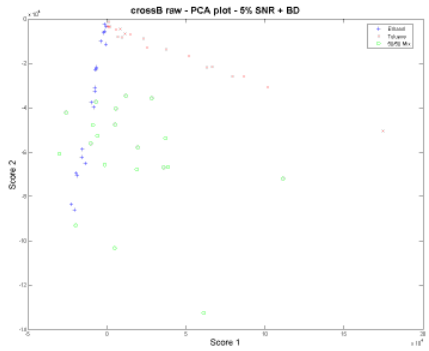
c-Nose Cross-convolution, BD



c-Nose Cross-convolution, SV + SNR



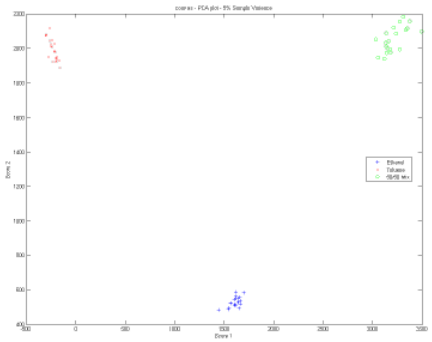
c-Nose Cross-convolution, SV + BD



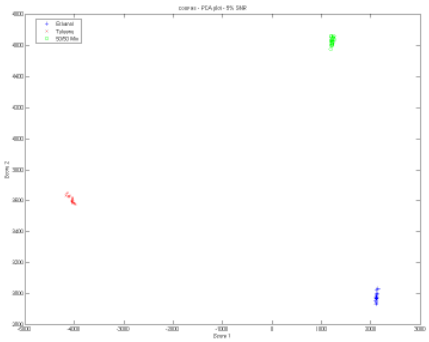
c-Nose Cross-convolution, SNR + BD

Appendix C: AOM Simulation PCA Results

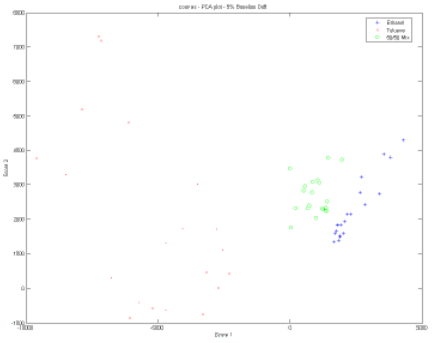
Autoscaling Pre-Processing



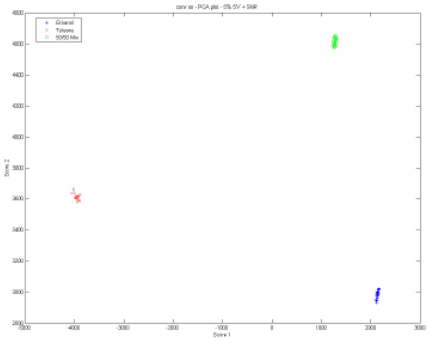
Convolution, Sample Variance



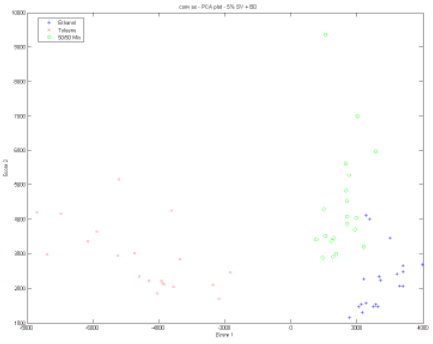
Convolution, Additive Noise



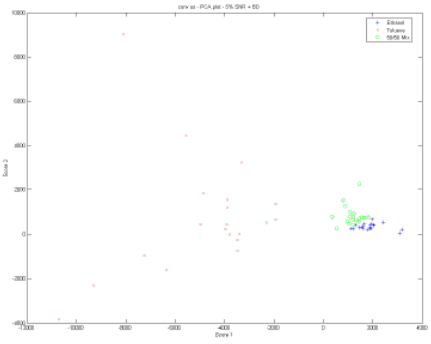
Convolution, Baseline Drift



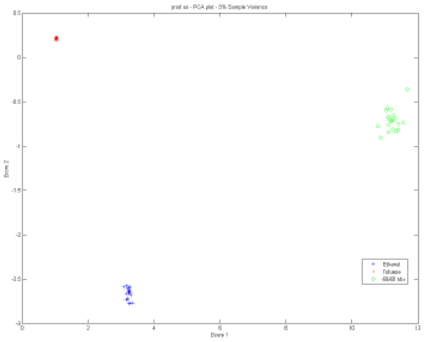
Convolution, SV + SNR



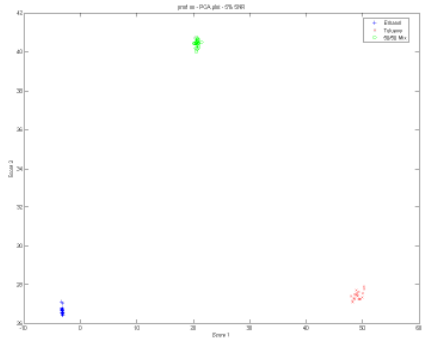
Convolution, SV + BD



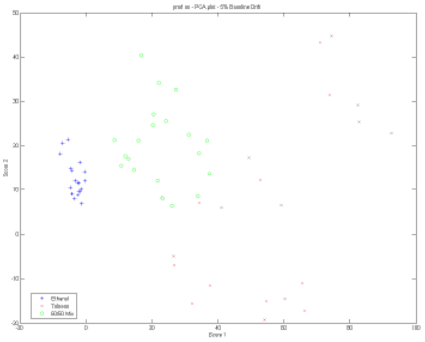
Convolution, SNR + BD



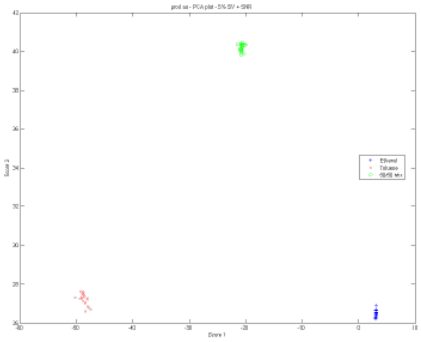
Product, SV



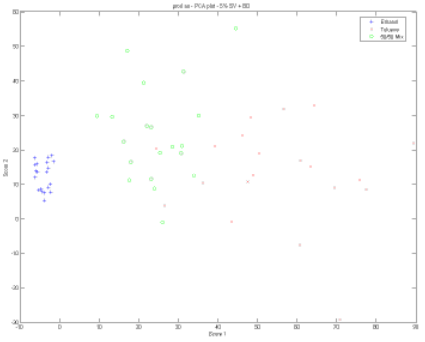
Product, SNR



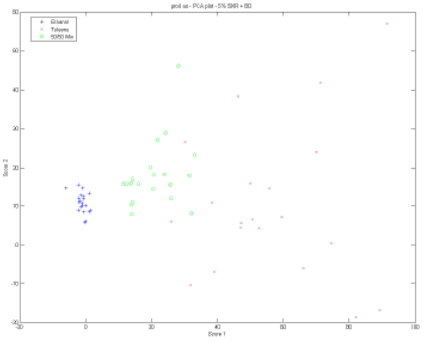
Product, BD



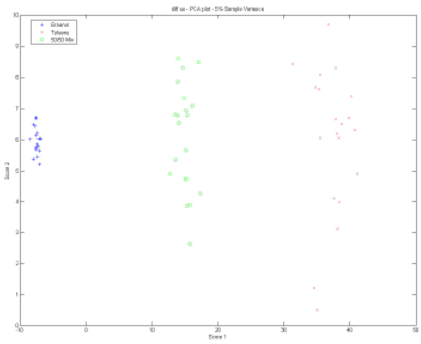
Product, SV + SNR



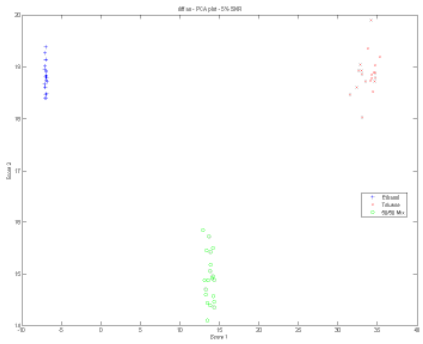
Product, SV + BD



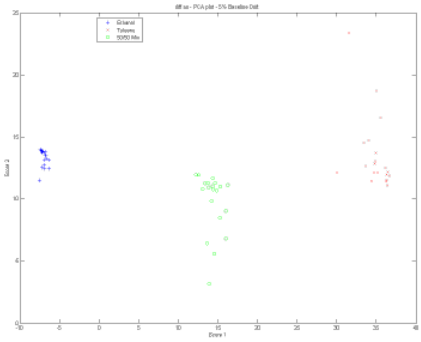
Product, SNR + BD



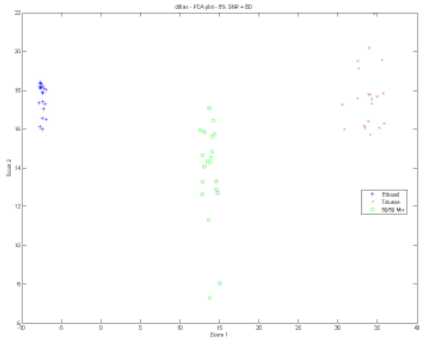
Difference, SV



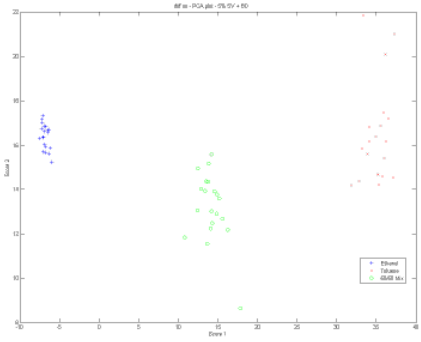
Difference, SNR



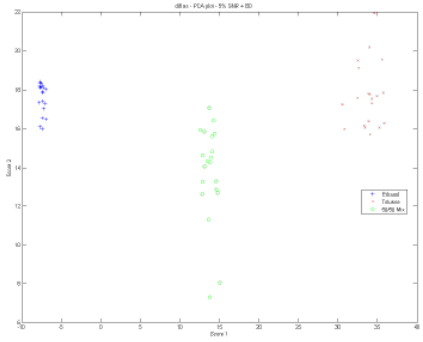
Difference, BD



Difference, SV + SNR

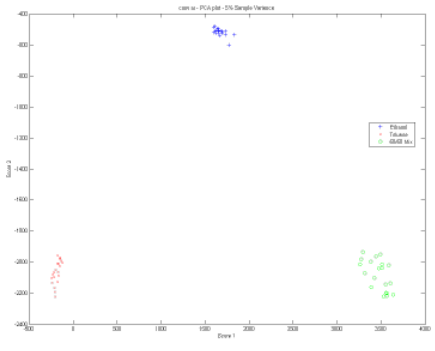


Difference, SV + BD

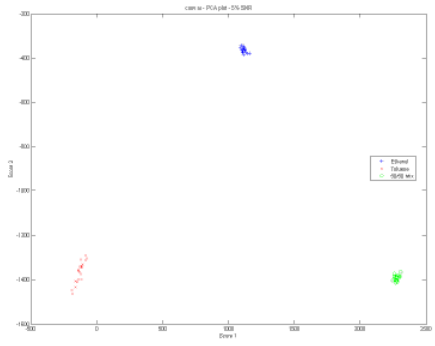


Difference, SNR + BD

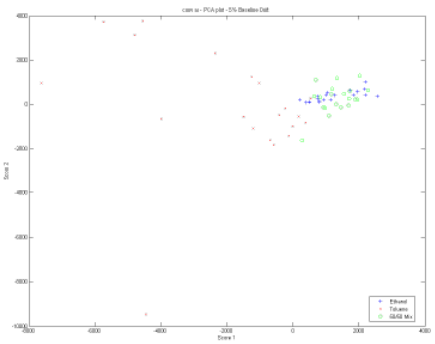
Autoranging Pre-processing



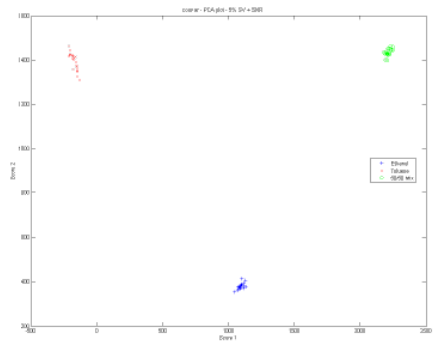
Convolution, SV



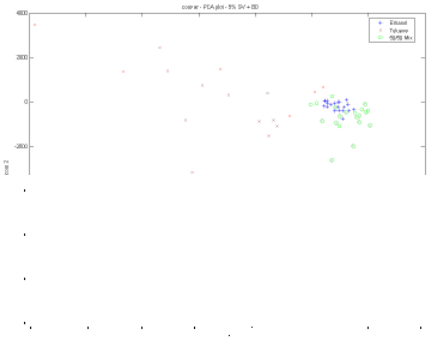
Convolution, SNR



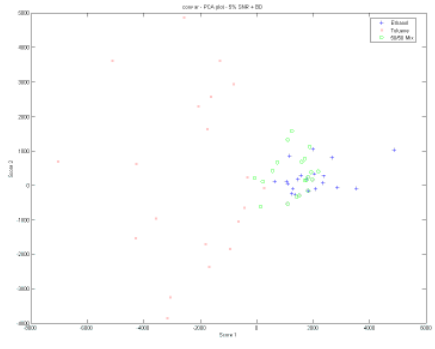
Convolution, BD



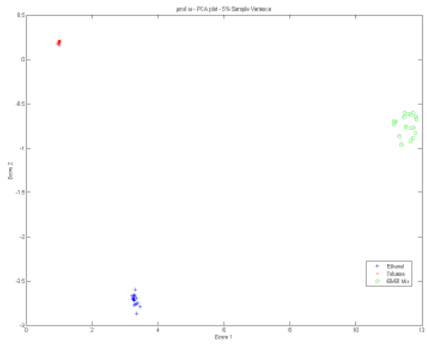
Convolution, SV + SNR



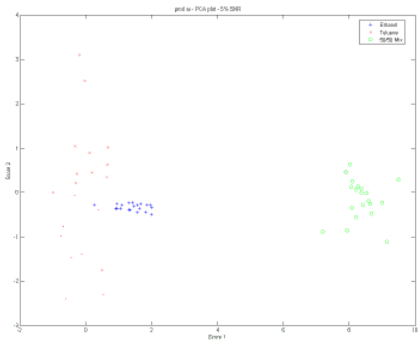
Convolution, SV + BD



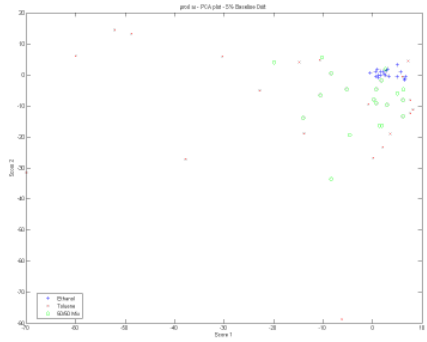
Convolution, SNR + BD



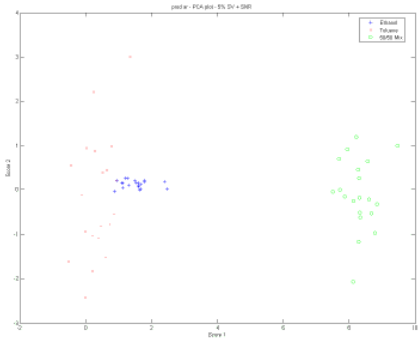
Product, SV



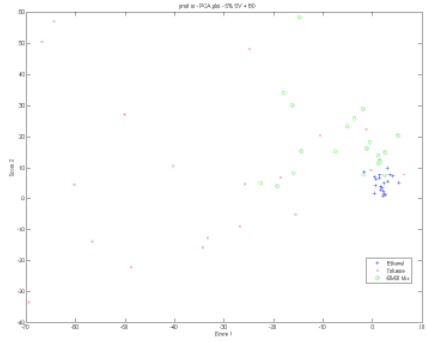
Product, SNR



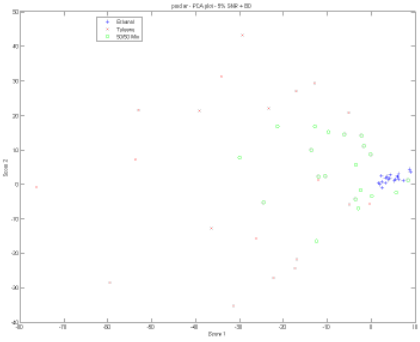
Product, BD



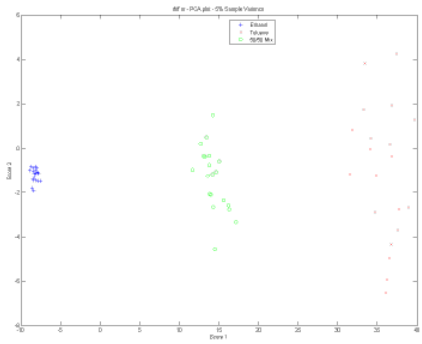
Product, SV + SNR



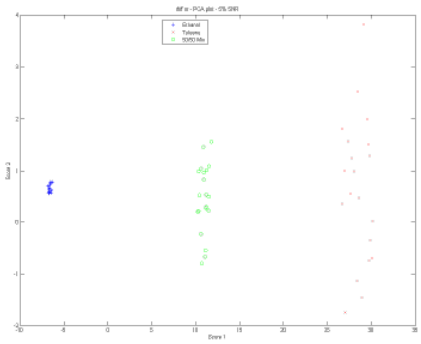
Product, SV + BD



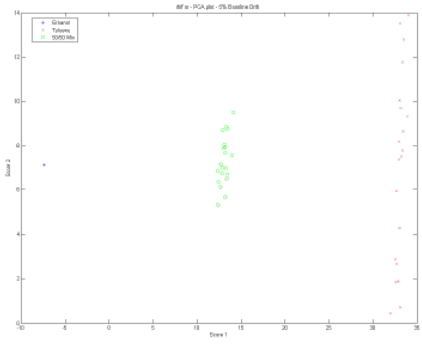
Product, SNR + BD



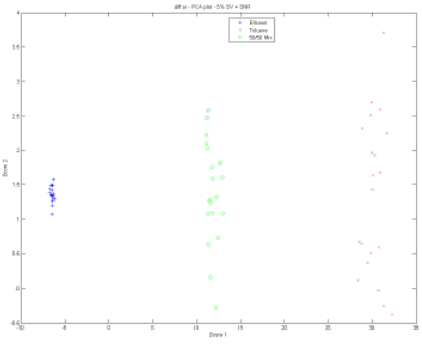
Difference, SV



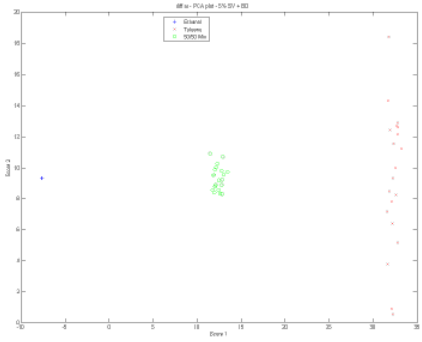
Difference, SNR



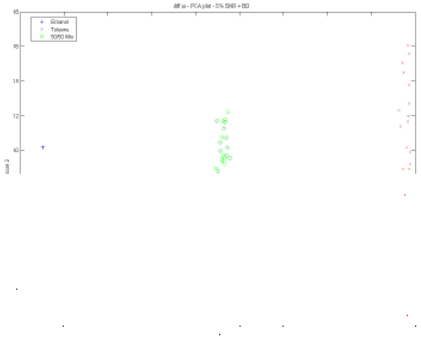
Difference, BD



Difference, SV + SNR

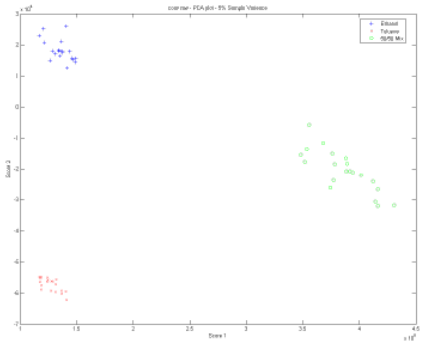


Difference, SV + BD

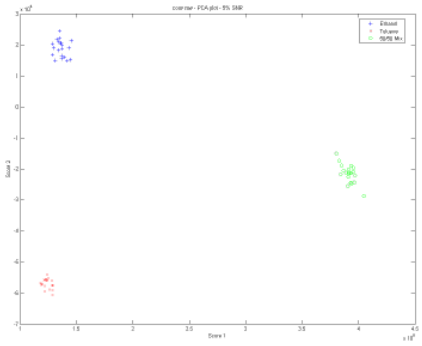


Difference, SNR + BD

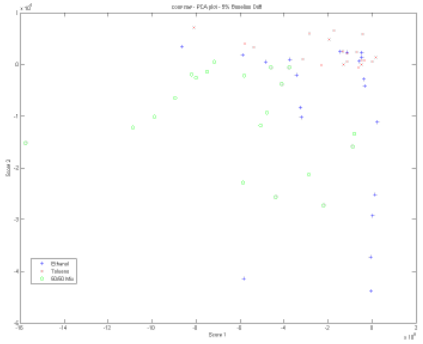
No Pre-processing



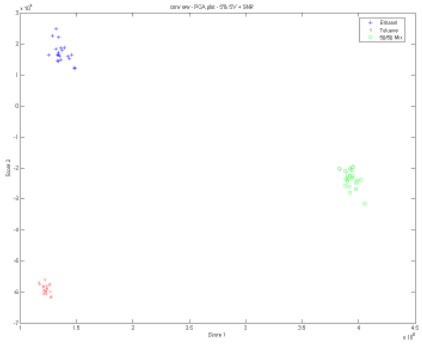
Convolution, SV



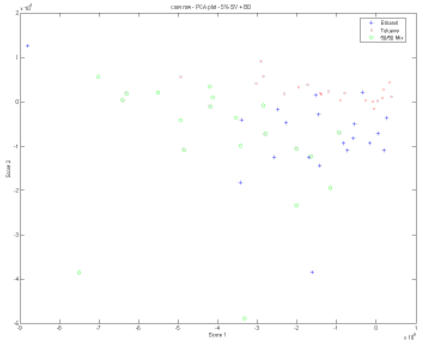
Convolution, SNR



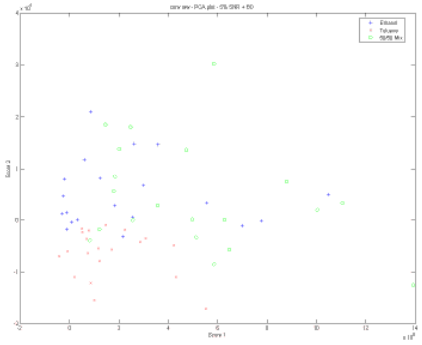
Convolution, BD



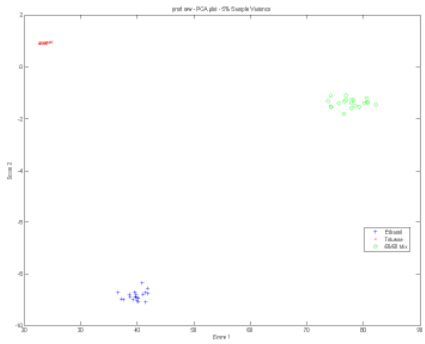
Convolution, SV + SNR



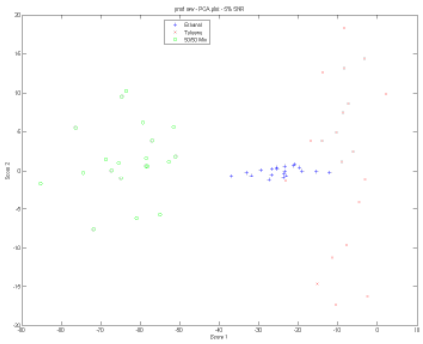
Convolution, SV + BD



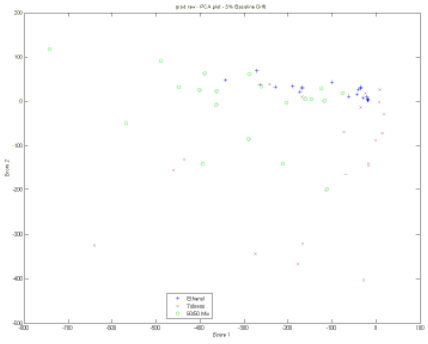
Convolution, SNR + BD



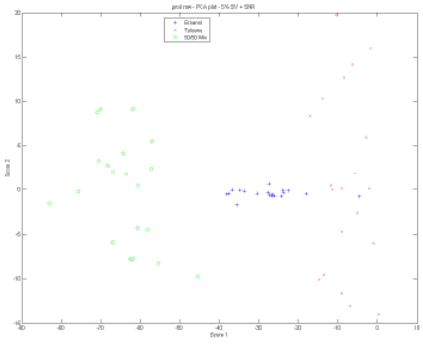
Product, SV



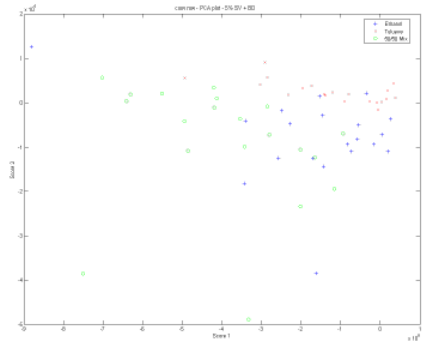
Product, SNR



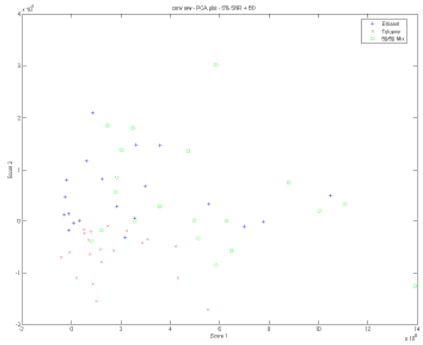
Product, BD



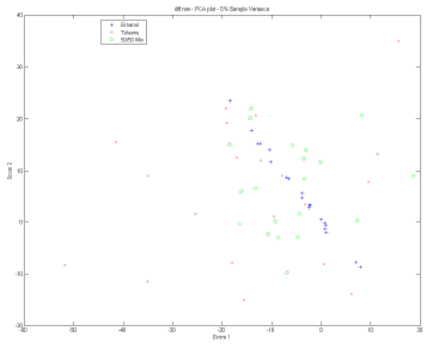
Product, SV + SNR



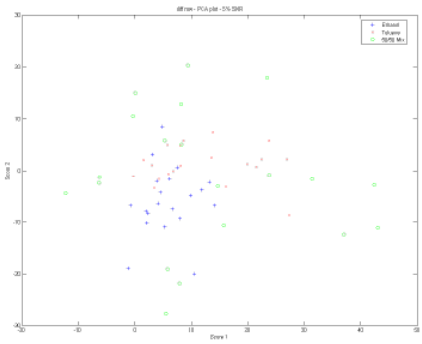
Product, SV + BD



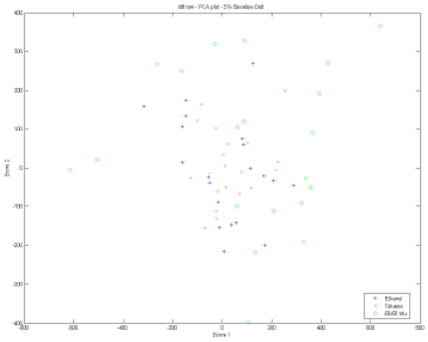
Product, SNR + BD



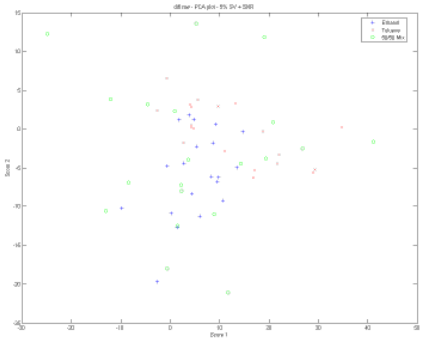
Difference, SV



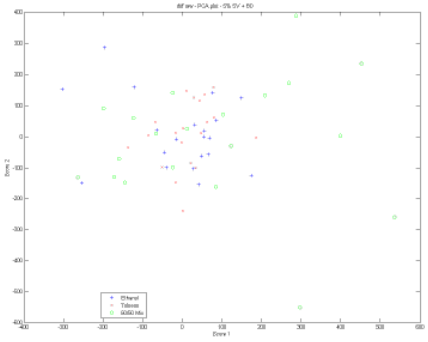
Difference, SNR



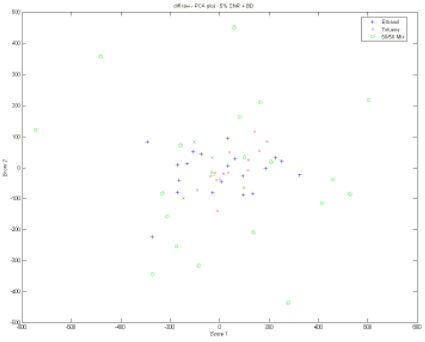
Difference, BD



Difference, SV + SNR



Difference, SV + BD



Difference, SNR + BD